JPL Contract 952532

A Study Program on the Development of a Mathematical Model(s) for Microbial Burden Prediction

Final Report
Addendum

Volume VI
Technical Report, Phase VII
Revised User's Manual for the Microbial Burden Prediction Model

OCTOBER 1969

MARTIN MARIETTA CORPORATION

As a result of work done in Phase VII (JPL Contract 952532), a new user's manual (Volume VI) has been prepared. Volume VI supercedes all user instructions in Volume II, but Volume II should still be used for explanations of model theory and meanings of the variables.

(Sheet to be pasted in Volume II, page ii)

MCR-68-97

A STUDY PROGRAM ON THE DEVELOPMENT OF MATHEMATICAL MODEL(S) FOR MICROBIAL BURDEN PREDICTION

JPL Contract 952532

Volume VI

Technical Report, Phase VII
Revised User's Manual

Prepared for:

Jet Propulsion Laboratory
4800 Oak Grove Drive
Pasadena, California 91103

Approved:

Bruce A. Nelson

Bruce A. Nelson
Program Manager

George E. Fosdick

George E. Fosdick
Chief
Operations Analysis

J. C. Curlander

J. C. Curlander
Manager
Systems Engineering

Martin Marietta Corporation
P. O. Box 179
Denver, Colorado 80201

FOREWORD

This document describes the work performed during Phase VII of JPL Contract 952532. (This is a follow-on to JPL Contract 952028). Phase VII accomplished improvements in the computer programs generated in Phase III.

This document also contains a revised User's Manual for the Microbial Burden Prediction Model and supplants the previous User's Manual (Volume II) for this purpose. Volume II is still useful for details of the theory of the model.

CONTENTS

ILLUSTRATIONS

DEFINITIONS

Alphameric - (Short for Alphabetic-numeric)

Data such as "PART 2W31" that may consist of a mixture of letters, numerals, and punctuation.

Card Image The data on a tape or disk that represents a punched card.

Character The contents of one column on a punched card or its equivalent on tape or disk.

Disk A storage device in a computer.

Field A group of columns on a punched card that is devoted to one quantity.

Floating Point

A number expressed as a characteristic and an exponent as in scientific notation: 3.2×10^4. When punched on cards or printed by the computer, the above number would appear as 3.2E+4. (See Integer.)

Histogram (Used interchangeably with "distribution" in this report). A method of representing a probability distribution by means of a number of intervals with a probability associated with each; a "bar graph" of the probability density.

Integer A computer quantity opposed to Floating Point. An integer or "fixed point" number repsents a whole number, has no decimal point, and must be as far right in its field as it can go (right justified) on a punched card.

Record See Tape

Stage, Task, Subtask, Operation - Terms used to denote the level of activity in an assembly and test sequence to be simulated by the burden prediction model. The operation is the most detailed level.

Definitions (Continued)

Tape — Magnetic tape, a storage medium used by computers. Information on tapes (or disks) consists of "logical records" each of which contains a number of computer words. Information is written on and read off tapes (or disks) by logical records.

Word — A storage location in a computer that can contain either a single number (integer or floating point) or several alphameric characters.

I. INTRODUCTION

During Phases I, II, and III of JPL Contract 952028, the requirements for a microbial burden prediction model were established, and the model was developed and programmed for computer. During Phases IV, V, and VI, the model was used to estimate the burden on the Mariner V Spacecraft. References 1 through 5 describe this work.

As a result of the above work, a number of changes were identified that would improve the effectiveness of the existing computer programs. These changes were grouped into four tasks (JPL Contract 952532) as described in the following sections.

A. Daily Restarts. This modification allows new data (e.g., the assembly and test activities performed during one day) to be incorporated to update a previous estimate. The results of the previous run are written on tape and saved to initialize the new run. This capability permits frequent updates without the wasted computer time involved in recomputing from the entire data deck. An advantage is that, if desired, a tape may be written for the last task successfully completed so that an abnormal job termination will not require recomputing the entire run.

In making this change, the original restart subroutine was removed and all "write" statements for Tape 11 (which prevously had recorded both restart and detailed data) were modified to contain only the detailed information not normally printed during a run. A data retrieval program was prepared to print information on selected tasks and subtasks from Tape 11.

B. Alteration of Data on Tape. The input data cards for the prediction program were previously read and checked by a data check program and then written on a tape (Tape 9). Any subsequent changes in data required changing the data cards and re-running the data check program. To avoid working with the card deck, a computer program was prepared to alter the data already on tape, including checking the resulting data and writing it on a new tape. Changes possible by this method consist of:

a) Changing any parameter on any card;

b) Adding new data (from cards) at any point in the tape;

c) Changing any parameter (either by replacing it or by multiplying it by a constant) every time it occurs between specified points on the tape;

d) Deleting specified card images from the tape.

The capability b) also permits data inputs entirely from cards in case no data tape is used.

As a data tape is prepared, a listing is made identifying each tape record by reference number (sequential), record type, stage, task, and subtask. The reference number is used for specifying future changes in that tape.

C. Faster Running. Burden calculations are performed using histograms, and the time required to perform a given run varies approximately as the square of the number of histogram intervals used. The capability was added to permit specifying the number of intervals (up to ten) to be used for each task. This allows a rough estimation to be run with one or two intervals (or even no

intervals; that is using mean values) using little computer time.

D. Print-out Options. The Mariner V runs took over two hours to print out the results. To reduce paper volume and print-out time, a more compact output format is now used. This output can be specified to be at the subtask, operation, or part level as desired for each task. Burden histograms are no longer printed except for inputs and task summaries (all burden histograms are recorded on Tape 11 for printing if desired), but some information has been included in the normal printout that was not printed before (e.g., the number of men working near the hardware).

During these four tasks, it was found useful to combine the programs for tape alteration (including card reading), burden estimation, and detailed printout from Tape 11 into one program so that separate submittals were not necessary. The combination requires more central memory storage in the computer than the individual programs do (unless segmentation or overlay is used). The three programs may easily be separated if the combination is not desired.

The programs are designed to run on the JPL Univac 1108 computer, but were programmed so that they could run on other large computers with essentially no changes.

The present volume (Volume VI) is complete as far as users of the computer programs are concerned; it supercedes all user instructions in the previous User's Manual (Volume II). However, Volume II contains explanations of the model and the meanings of the variables used. Volume II should therefore be regarded as a reference rather than a user's manual.

II. TECHNICAL DISCUSSION (Revised User's Manual)

A. Descriptions of Changes Made

1. Daily Restarts. To provide an improved restart capability, a dump is made (on Tape 2) of all information "remembered" in the computer from one control card to the next. This information is approximately that contained in blank common storage, but differs to the extent that a few extra variables are included and any unused histograms are omitted. The tape records are in binary, and will vary in number 100 to 1000 depending on the number of histograms used.

Three options are provided for this feature:

1) Restart dump at end of run only;
2) Restart dump at the end of the last task successfully completed;
3) No restart dumps.

These options are selected by setting the variable I2 on any control card and can be changed during the run if desired. (See Section IIC for card preparation.) Option #2 preserves computed results task by task so that an error exit from the program will require recomputation only from last completed task instead of re-running the entire data deck.

The burden prediction program previously included a restart capability, embodied in subroutine MBRS, the Microbial Burden Restart Subroutine, which

made use of detailed information recorded on Tape 11 during a previous run. Subroutine MBRS is no longer used, and Tape 11 is now used only for recording detailed burden information which is not normally of interest and would thus cause unnecessary printout volume (and time) if included in the printed output of the burden prediction program.

All records written on Tape 11 are identical, consisting of 43 words. In general, each record contains one histogram, the run, stage, task, and subtask numbers, the subtask description, and ten additional parameters. Each record is identified as to type, so that the information recorded in each word can be identified and printed. A separate computer program (subroutine DPS, PLS, and HPS) has been written to locate and print out selected tasks and subtasks from Tape 11. The information to be printed is determined by a set of instruction cards, which can specify either a task (or subtask) to be printed or that all information between a given task (or subtask) and another. Any number of such instruction cards may be used. Specification of the subtask is optional; if not supplied, selection will be by tasks. See Table 1 for typical output from this program. Table 2 describes the instruction types.

To facilitate frequent updates in burden prediction, provision has been made for data inputs to the prediction program from cards instead of tape. The card data were previously processed by a Data Check Program and written on tape (Tape 9) for later use by the prediction program, thus requiring two job submittals to obtain a burden prediction. The functions of data checking and recording (now accomplished by the Tape Alteration Subroutine; see Section II.A.2) have been combined in one program with the burden prediction program and the

```
 RANGE =    1.58E-03  9.04E+02


PART BURDEN AFTER UPDATE                RUN 1, STAGE  7, TASK   5, SUBTASK  8  INST BOOST DAMPERS
PART   5  SQX-3762-599, SURF. 1, AREA=    .50, TIME=  80.000, ENVIRON. -0                 BURDEN MEAN VALUE =  4.52129E+02
 PROBABILITY =          1.00000
 RANGE =    1.58E-03  9.04E+02


PART BURDEN AFTER DECONTAMINATION       RUN 1, STAGE  7, TASK   5, SUBTASK  8  INST BOOST DAMPERS
PART   5, SURF. 1, AREA=    .50, OPER.  1,   .50 HOURS, FRACTION REMOVED=  .500           BURDEN MEAN VALUE =  2.26064E+02
 PROBABILITY =          1.00000
 RANGE =    3.94E-04  4.52E+02


PART BURDEN AFTER FALLOUT               RUN 1, STAGE  7, TASK   5, SUBTASK  8  INST BOOST DAMPERS
PART   5, SURF. 1, AREA=    .50, OPER.  1, 10.00 HOURS, ENVIRON.  1,   1 MEN.             BURDEN MEAN VALUE =  4.60786E+02
 PROBABILITY =          1.00000
 RANGE =    2.20E-04  9.22E+02


PART BURDEN AFTER CONTACT               RUN 1, STAGE  7, TASK   5, SUBTASK  8  INST BOOST DAMPERS
PART   5, SURF. 1, AREA=    .50, CONTACT= .100, S1=7.000, S2=1.000, TOOL=    1.0          BURDEN MEAN VALUE =  4.15057E+02
 PROBABILITY =          1.00000
 RANGE =    2.10E-04  8.30E+02


---- START OF NEW SUBTASK ----          RUN 1, STAGE  7, TASK  99, SUBTASK  1  INSERT GUIDE PIN
PREREQUISITE SUBTASKS-  -0,  -0                                                       START TIME MEAN VALUE =  2.64905E+02
 PROBABILITY =          1.00000
 RANGE =    3.01E+01  5.00E+02


PART BURDEN AFTER UPDATE                RUN 1, STAGE  7, TASK  99, SUBTASK  1  INSERT GUIDE PIN
PART   1  SQX-3755-420, SURF. 1, AREA=    .10, TIME= 264.905, ENVIRON. -0                 BURDEN MEAN VALUE =  3.69665E+02
 PROBABILITY =          1.00000
 RANGE =    6.09E-10  7.39E+02


PART BURDEN AFTER FALLOUT               RUN 1, STAGE  7, TASK  99, SUBTASK  1  INSERT GUIDE PIN
PART   1, SURF. 1, AREA=    .10, OPER.  1, 10.00 HOURS, ENVIRON.  1,   1 MEN.             BURDEN MEAN VALUE =  3.66502E+02
 PROBABILITY =           .10000    .20000    .30000    .15000    .10000    .05000    .04000    .03000    .02000    .01000
 RANGE =    3.58E-08  2.13E+01  6.40E+01  1.30E+02  2.16E+02  6.09E+02  8.05E+02  9.62E+02  1.08E+03  1.16E+03  2.62E+04


PART BURDEN AFTER UPDATE                RUN 1, STAGE  7, TASK  99, SUBTASK  1  INSERT GUIDE PIN
PART   5  SQX-3762-599, SURF. 1, AREA=    .50, TIME= 274.905, ENVIRON. -0                 BURDEN MEAN VALUE =  4.15057E+02
 PROBABILITY =          1.00000
 RANGE =    2.10E-04  8.30E+02


PART BURDEN AFTER FALLOUT               RUN 1, STAGE  7, TASK  99, SUBTASK  1  INSERT GUIDE PIN
PART   5, SURF. 1, AREA=    .50, OPER.  1, 10.00 HOURS, ENVIRON.  1,   3 MEN.             BURDEN MEAN VALUE =  7.52761E+02
 PROBABILITY =           .10000    .20000    .30000    .15000    .10000    .05000    .04000    .03000    .02000    .01000
 RANGE =    1.82E-07  9.66E+00  2.75E+01  5.41E+01  1.20E+02  1.97E+02  2.36E+02  2.67E+02  2.90E+02  8.98E+02  1.32E+05


PART BURDEN AFTER CONTACT               RUN 1, STAGE  7, TASK  99, SUBTASK  1  INSERT GUIDE PIN
PART   5, SURF. 1, AREA=    .50, CONTACT= .100, S1=7.000, S2=1.000, TOOL=    1.0          BURDEN MEAN VALUE =  6.77835E+02
 PROBABILITY =           .10000    .20000    .30000    .15000    .10000    .05000    .04000    .03000    .02000    .01000
 RANGE =    7.69E-06  5.47E+00  2.15E+01  4.64E+01  1.07E+02  1.81E+02  2.15E+02  2.51E+02  2.65E+02  5.20E+02  1.20E+05
```

Table 1. Typical Output, Detailed Printout Program

Table 2 Detailed Printout Instructions

Instruction Format:

Columns:	5	9-10	13-15	19-20	24-25	28-30	34-35
	K	NS	NT	NST	NS2	NT2	NST2

All numbers must be right justified in their fields.

K determines the printout:

K = 0: No more printing (program exit).

K = 1: Print all information on stage NS, task NT, Subtask NST

K = 2: Print all information from stage NS, task NT, subtask NST to (but not including) stage NS2, task NT2, subtask NST2.

The subtask specification is optional; if omitted, the entire task will be printed. If all information beyond a specified task is desired, use a card with K = 2 and NS2 blank.

program to print from Tape 11 so that multiple job submittals are avoided. This was accomplished by making subroutines of the three programs and writing one main program that does nothing but call the three sub-programs in turn. If all three are not desired, the main program is easily altered (by removing appropriate "CALL..." cards) to accomplish this. The data interfaces between these sub-programs (Tape 9 between card reading and burden prediction, and Tape 11 between burden prediction and detailed printout) can be disc files unless permanent records are desired on tape; this option is determined by the presence or absence of tape request cards in the job control deck, the disc being used if a tape is not supplied.

2. Alteration of Data on Tape. To alter a parameter on a card record on tape, three items of information are necessary:

1) which tape record;
2) which parameter on that record;
3) the new value of the parameter.

Tape records were constructed that, with one exception, are of uniform length: seven integer words plus seven floating point or alphameric words. The exception is the histogram record, which consists of one integer plus 22 floating point words. The first word of each record identifies the record type and hence, the information contained in it. Table 3 shows the information layout of these records, and also shows the numbering system (from 1 through 13 except for histogram records, which are 1 through 22) used to identify the parameter to be changed.

Table 3 Tape 9 Record Formats

Card Type	PARAMETERS												
	INTEGER						FLOATING OR ALPHAMERIC						
	1	2	3	4	5	6	7	8	9	10	11	12	13
CC	KK	IK	I1	I2	I5	I6	I3	I4					
RD	KR						← RUN (7) →						
SD	KS						← STG (7) →						
TD	KT	L1	L2	L3	L4		← TSK (7) →						
EM							AES(1)	AES(2)	AES(3)	AES(4)			
ED	N						← DSC(4) →						
EQ	IEC	IET					AEC	AET	AED	AEF(1)	AEF(2)	AEF(3)	AEF(4)
OD	N						← DSC(4) →						
OQ	IOT	IOQ	IOC										
PD	N						← DSC(6) →						
PQ	IAB(1)	IAB(2)	IAB(3)	IAB(4)	IAB(5)	IAB(6)	AAG	AAS					
DD	N	M	L				← DSC(4) →						
KD	N	N1	N2				← DSC(4) →						
KC	K	IR(1)	IR(2)	IR(3)	IR(4)		AR(1)	AR(2)	AR(3)				
KO	IO	IKE					AKT	AKQ					
PE	IP	LS					APD	APC	APS	APA(1)	APA(2)		
PE	IP	LK					AR(1)	AR(2)	AR(3)	AR(4)			
ZD	IZ						← DSC(4) →						
ZC	IP						FP(1)	FP(2)	FP(3)	FP(4)			
DQ	Different Format 22 Parameters: DR(N,1), DR(N,2),...,DR(N,11), XR(N,1)...,XR(N,11) Numbered 1 thru 22.												

Identification of the tape record containing the desired parameter is most conveniently done by specifying the stage, task, subtask, etc. in the data deck since this is the way the data is organized. However duplicate task and subtask numbers are permitted, so such identification is not unique. For this and other reasons, the primary means of identifying the tape record to be altered is a single number, the number of that record in a sequential listing of all records on the tape.

When a data tape (Tape 9) is written, a listing is made of the data on each record in the format of Table 3. In addition to the record type and parameter values, the record number and the stage, task and subtask are printed. A typical listing is given in Table 4.

The types of changes possible with this program are listed in Table 5. Essentially, any change that could be made in a card deck can be made in the tape records; the main difference is that it is not possible to alter just one character in tape record - the entire parameter must be replaced (or in the case of alphameric fields, a new card must be read to replace the record to be altered).

When this program is used, the old data tape (if any) is designated Tape 12 and a new Tape 9 is written on a blank tape. A card deck must be provided to specify the changes to be made and to provide any new data cards to be added. Note that Tape 12 is not necessary; input can be entirely from cards if desired.

Table 4. Typical Output, Tape Alteration Program

```
RECORD, TYPE, S/TSK/ST    1   2   3   4   5   6      7          8          9         10         11         12         13

----- CHANGE          0     2 9999       -0.

   1   CC                 1   1   3   3  12   1
   2   RD                 1                          MARINER V ASSEMBLY AND TEST
   3   SD    1/           1                          SPACECRAFT ASSEMBLY
   4   TD    1/   6/      6  -0  -0  12  -0          INSTALL BAY V SOLAR PANEL
   5   EM    1/   6/                                 1.000      1.000      1.000      1.000
   6   ED    1/   6/      1                          LAMINAR FLOW TENT
   7   EQ    1/   6/      1   1                       .100      50.000     1.000     60.000     60.000     60.000     60.000
   8   ED    1/   6/      0
   9   OD    1/   6/      1                          VISUAL INSP.
  10   OQ    1/   6/      1   1   1
  11   OD    1/   6/      2                          MANIPULATE BY HAND
  12   OQ    1/   6/      1   1   1
  13   OD    1/   6/      0
  14   PD    1/   6/      1                          SOLAR PANEL, BAY V      SQX-3755-420
  15   PQ    1/   6/     22  -0  -0  -0  -0   1      1.000      1.000
  16   PD    1/   6/      0
  17   DD    1/   6/     12  11                      TEN BUMP ID HISTOGRAM
  18   DQ    1/   6/
            0.      .100000    .200000    .300000    .150000    .100000    .050000    .040000    .030000    .020000    .010000
       -0.       -0.        -0.        -0.        -0.        -0.        -0.        -0.        -0.        -0.        -0.
  19   DD    1/   6/      1  11                      BASIC DISTRIBUTION
  20   DQ    1/   6/
            0.      .100000    .200000    .300000    .150000    .100000    .050000    .040000    .030000    .020000    .010000
       1.00E+00  2.00E+00   5.00E+00   1.00E+01   2.00E+01   5.00E+01   1.00E+02   2.00E+02   5.00E+02   1.00E+03   2.00E+03
  21   DD    1/   6/     22   6                      PART BURDEN SQX3755-420
  22   DQ    1/   6/
            .10     .200000    .400000    .200000    .150000    .049900

******ERROR    1  TOTAL PROB. NOT EQUAL TO 1

       1.00E+00  3.00E+00   1.00E+01   3.00E+01   1.00E+02   3.00E+02
  23   DD    1/   6/      0  -0
  24   KD    1/   6/ 1    1  -0  -0                   PREP OF SOLAR PANEL
  25   KC    1/   6/ 1    0  -0  -0  -0  -0  -0      -0.        -0.        -0.
  26   KO    1/   6/ 1    1  -0                      10.000     1.000
  27   PE    1/   6/ 1    1  -0                      -0.        -0.        -0.        -0.        -0.
                                                        PART    1 SURFACE AREAS=      .100       0.         0.         0.
  28   PE    1/   6/ 1    0  -0                      -0.        -0.        -0.        -0.        -0.
  29   KO    1/   6/ 1    1  -0                      10.000     3.000
  30   PE    1/   6/ 1    1  -0                      -0.        -0.        -0.        -0.        -0.
                                                        PART    1 SURFACE AREAS=      .100       0.         0.         0.
  31   PE    1/   6/ 1    0  -0                      -0.        -0.        -0.        -0.        -0.
  32   KO    1/   6/ 1    1  -0                      10.000     2.000
  33   PE    1/   6/ 1    1  -0                      -0.        -0.        -0.        -0.        -0.
                                                        PART    1 SURFACE AREAS=      .100       0.         0.         0.
  34   PE    1/   6/ 1    0  -0                      -0.        -0.        -0.        -0.        -0.
  35   KO    1/   6/ 1    1  -0                      10.000     3.000
  36   PE    1/   6/ 1    1  -0                      -0.        -0.        -0.        -0.        -0.
                                                        PART    1 SURFACE AREAS=      .100       0.         0.         0.
  37   PE    1/   6/ 1    0  -0                      -0.        -0.        -0.        -0.        -0.
  38   KO    1/   6/ 1    1  -0                      10.000     1.000
```

Table 5 Tape 9 Alternation Instructions

Instruction Format:

Columns:	1-10*	15	20	21-30*	35
	N	K	M	F	J

*Integers in these fields must be right justified.

The value of J controls the error message printout:

J = 0 : No change. (J is initially set to 1.)

J = 1 : Print fatal errors only.

J = 2 : Print all errors.

K determines the type of change to be made:

K = 0 : Ignore all tape records after record N.

K = 1 : Replace record N with a new card

K = 2 : Add M cards after record N.

K = 3 : Delete M records starting with record N.

K = 4 : Replace parameter M on record N by F.

K = 5 : Multiply parameter M on record N by F.

K = 6 : Begin replacing parameter M by F starting with record N.

K = 7 : Begin multiplying parameter M by F starting with record N.

K = 8 : Stop altering parameter M. (Record N is the first record not to be altered.)

NOTES:

1) Instructions must be in order of record number (N);

2) No changes are made in parameters on cards read in;

3) An instruction with K=0, 1, or 2 must be the last for that record number;

4) Cards to be added must be placed directly behind the instruction;

5) No re-start control card may appear in data on Tape 9; restarts must be on cards only.

6) To read from cards only (no previous data tape), use an instruction card with N=0, K=2, and M=99999 (or any number larger than the number of cards to be read).

7) A K=0 instruction card should be used whenever only part of Tape 12 is to be used; merely setting KK=0 on a control card is not sufficient.

As the data from Tape 12 and from cards is combined to form the new data tape, the records are checked for errors such as missing data or data out of range. This data checking also maintains the current surface areas of each affected part, and these are printed each time the corresponding part is mentioned. Whenever an error occurs, an error message is printed, and a count is kept of the number of errors of each type. (See Table 6 for the listing of error types.) Errors are either fatal (e.g., card out of sequence) or non-fatal (e.g., area of contact is greater than surface area). Fatal errors are those that will cause an error exit from the burden prediction program, whereas non-fatal errors will not stop the prediction program but do indicate a problem in the data. When the tape alteration program is combined with the prediction program, any fatal errors in the data will terminate the job before running the prediction program.

An advantage of the uniform tape format is that an out-of-sequence record will seldom cause termination of the processing; a new Tape 9 will be written in spite of errors. Two exceptions are the non-uniform histogram record and any data read in from cards (which do not have uniform formats and are not identified as to record type).

Since some data tapes exist that were written in a different format (essentially card images), an extra subroutine is provided for converting such tapes to the new format. This is necessary since the burden prediction program has been modified to accept the new tape record formats and can no longer use the old data tapes.

Table 6 Data Check Error Types

1) Index out of range;
2) Prerequisite missing;
3) Necessary data missing;
4) Too many cards in set;
5) Index used previously;
6) Data out of range;
7) Card out of sequence;
8) Affected part area zero;
9) Contact area greater than surface area
10) Remaining area negative;
11) Total probability not equal to 1.

Note: Errors 1 through 7 are fatal;
Errors 8 through 11 are warnings.

3. CONTROL OF THE NUMBER OF HISTOGRAM INTERVALS

The maximum number of intervals to be used in histogram calculations is adjustable in the range from zero (i.e., a constant) through ten. This is accomplished by specifying as I5 on any control card (CC) the identification number of a histogram having the desired number of intervals. This "model" histogram also provides the probability levels to which burden histograms are adjusted. (See Ref. 4, p 35-36 for discussion of the need for this.)

Subroutine HAS, the Histogram Adjusting Subroutine, was written to alter the number of intervals and the probability levels of any specified histogram to match the number of intervals in the model histogram. This is done by constructing the cumulative probability function F(x) of the histogram to be altered and determining by interpolation the value of x for which F(x) equals each desired probability level. (A numerical example is given in Ref. 4, p 50-51.) This routine was originally programmed as a part of HCS, the Histogram Combining Subroutine, and was separated and modified to serve in the present task.

Although the burden prediction program will run without having any model histogram specified, it is best to provide one. Assuming that this has been done, all input histograms are checked to see if they have too many intervals, and if so, they are adjusted by subroutine HAS. One exception to this is that the general purpose distributions (as opposed to histograms representing burdens on parts) are preserved in their original form in separate storage locations before they are adjusted by HAS. This permits these histograms, which must have identification numbers in the range 1 through 20, to be reconstituted without loss of accuracy if a larger number of histogram intervals is specified later in the run.

Whenever a new model histogram is specified, the first 20 histograms are retrieved from their storage locations in their original forms and adjusted by subroutine HAS. If the new model histogram has an equal or larger number of intervals than the previous one, this is all that is done. However if the new model histogram has fewer intervals, all stored histograms are checked and are adjusted if they have too many intervals.

Note that if I5 is changed on a control card to specify a new model histogram and if that histogram has not yet been read in, it must be the first histogram to be read in since it must be used to adjust all subsequent distributions.

4. PRINT OUT OPTIONS

The essence of this task was a reformatting of the printout of the burden prediction program to conserve space. A notable modification is that all histograms are now printed on three lines regardless of the number of intervals they contain. Many histograms that formerly were printed are now available only on TAPE 11 since their information was rarely used. Some compression of printout such as the Task Summary was made, but these still remain rather spread out for ease of reading.

Optional printout at the operation and part level is available, and is somewhat crowded when printed, being confined to one line per operation or part. (See Table 7 for a sample of this printout.) The printout option is controlled by setting parameter I1 on any control card (see page 28) and may be changed as desired.

```
STAGE  7  S/C ASSEMBLY + TEST, PASADENA
TASK  99  ALIGNMENT CHECKS


SUBTASK  1, INSERT GUIDE PIN
    OPERATION  1, ENVIRONMENT  1, FROM 264.91 TO 274.91 HOURS,  1 MEN
         PART   1  SQX-3755-420, SURFACE 1, AREA=    .100, TOOL BURDEN=    -0, AREA TOUCHED=-0.     , BURDEN= 3.665E+02
    OPERATION  1, ENVIRONMENT  1, FROM 274.91 TO 284.91 HOURS,  3 MEN
         PART   5  SQX-3762-599, SURFACE 1, AREA=    .500, TOOL BURDEN=     1, AREA TOUCHED=  .100, BURDEN= 7.528E+02
         PART   1  SQX-3755-420, SURFACE 1, AREA=    .100, TOOL BURDEN=    -0, AREA TOUCHED=-0.     , BURDEN= 4.114E+02
    OPERATION  1, ENVIRONMENT  1, FROM 284.91 TO 294.91 HOURS,  2 MEN
         PART   1  SQX-3755-420, SURFACE 1, AREA=    .100, TOOL BURDEN=    -0, AREA TOUCHED=-0.     , BURDEN= 4.289E+02
    OPERATION  1, ENVIRONMENT  1, FROM 294.91 TO 304.91 HOURS,  3 MEN
         PART   1  SQX-3755-420, SURFACE 1, AREA=    .100, TOOL BURDEN=    -0, AREA TOUCHED=-0.     , BURDEN= 4.688E+02
    OPERATION  1, ENVIRONMENT  1, FROM 304.91 TO 314.91 HOURS,  1 MEN
         PART   5  SQX-3762-599, SURFACE 1, AREA=    .500, TOOL BURDEN=     1, AREA TOUCHED=  .100, BURDEN= 7.554E+02
         PART   1  SQX-3755-420, SURFACE 1, AREA=    .100, TOOL BURDEN=    -0, AREA TOUCHED=-0.     , BURDEN= 4.578E+02

SUBTASK  2, INSTALL GAGE COLLAR
    OPERATION  1, ENVIRONMENT  1, FROM 264.91 TO 274.91 HOURS,  1 MEN
         PART   1  SQX-3755-420, SURFACE 1, AREA=    .100, TOOL BURDEN=    -0, AREA TOUCHED=-0.     , BURDEN= 4.476E+02

SUBTASK  3, INST LATCH ASSEMBLY
    OPERATION  1, ENVIRONMENT  1, FROM 404.93 TO 414.93 HOURS,  2 MEN
         PART   1  SQX-3755-420, SURFACE 1, AREA=    .100, TOOL BURDEN=    -0, AREA TOUCHED=-0.     , BURDEN= 4.621E+02
         PART   5  SQX-3762-599, SURFACE 1, AREA=    .500, TOOL BURDEN=     1, AREA TOUCHED=  .100, BURDEN= 8.771E+02
    OPERATION  1, ENVIRONMENT  1, FROM 414.93 TO 424.93 HOURS,  3 MEN
         PART   1  SQX-3755-420, SURFACE 1, AREA=    .100, TOOL BURDEN=    -0, AREA TOUCHED=-0.     , BURDEN= 4.994E+02
         PART   5  SQX-3762-599, SURFACE 1, AREA=    .500, TOOL BURDEN=     1, AREA TOUCHED=  .100, BURDEN= 1.098E+03
    OPERATION  1, ENVIRONMENT  1, FROM 424.93 TO 434.93 HOURS,  3 MEN
         PART   1  SQX-3755-420, SURFACE 1, AREA=    .100, TOOL BURDEN=    -0, AREA TOUCHED=-0.     , BURDEN= 5.338E+02
    OPERATION  1, ENVIRONMENT  1, FROM 434.93 TO 444.93 HOURS,  3 MEN
         PART   1  SQX-3755-420, SURFACE 1, AREA=    .100, TOOL BURDEN=    -0, AREA TOUCHED=-0.     , BURDEN= 5.654E+02

SUBTASK  5, ADJUST LATCH ASSY
    OPERATION  1, ENVIRONMENT  1, FROM 498.84 TO 508.84 HOURS,  3 MEN
         PART   1  SQX-3755-420, SURFACE 1, AREA=    .100, TOOL BURDEN=    -0, AREA TOUCHED=-0.     , BURDEN= 5.944E+02
    OPERATION  1, ENVIRONMENT  1, FROM 508.84 TO 518.84 HOURS,  1 MEN
         PART   5  SQX-3762-599, SURFACE 1, AREA=    .500, TOOL BURDEN=     1, AREA TOUCHED=  .100, BURDEN= 1.041E+03
         PART   1  SQX-3755-420, SURFACE 1, AREA=    .100, TOOL BURDEN=    -0, AREA TOUCHED=-0.     , BURDEN= 5.733E+02

SUBTASK  9, POSIT LATCH ASSY/JIG
    OPERATION  1, ENVIRONMENT  1, FROM 444.93 TO 454.93 HOURS,  1 MEN
         PART   1  SQX-3755-420, SURFACE 1, AREA=    .100, TOOL BURDEN=    -0, AREA TOUCHED=-0.     , BURDEN= 5.539E+02
    OPERATION  1, ENVIRONMENT  1, FROM 454.93 TO 464.93 HOURS,  1 MEN
         PART   1  SQX-3755-420, SURFACE 1, AREA=    .100, TOOL BURDEN=    -0, AREA TOUCHED=-0.     , BURDEN= 5.361E+02
    OPERATION  1, ENVIRONMENT  1, FROM 464.93 TO 474.93 HOURS,  1 MEN
         PART   1  SQX-3755-420, SURFACE 1, AREA=    .100, TOOL BURDEN=    -0, AREA TOUCHED=-0.     , BURDEN= 5.197E+02
    OPERATION  1, ENVIRONMENT  1, FROM 474.93 TO 484.93 HOURS,  1 MEN
         PART   1  SQX-3755-420, SURFACE 1, AREA=    .100, TOOL BURDEN=    -0, AREA TOUCHED=-0.     , BURDEN= 5.045E+02

SUBTASK 19, REM DAMPER FROM JIG
    OPERATION  1, ENVIRONMENT  1, FROM 484.93 TO 494.93 HOURS,  1 MEN
         PART   1  SQX-3755-420, SURFACE 1, AREA=    .100, TOOL BURDEN=    -0, AREA TOUCHED=-0.     , BURDEN= 4.906E+02
    OPERATION  1, ENVIRONMENT  1, FROM 494.93 TO 504.93 HOURS,  1 MEN
         PART   1  SQX-3755-420, SURFACE 1, AREA=    .100, TOOL BURDEN=    -0, AREA TOUCHED=-0.     , BURDEN= 4.778E+02
    OPERATION  1, ENVIRONMENT  1, FROM 504.93 TO 514.93 HOURS,  1 MEN
         PART   1  SQX-3755-420, SURFACE 1, AREA=    .100, TOOL BURDEN=    -0, AREA TOUCHED=-0.     , BURDEN= 4.660E+02
    OPERATION  1, ENVIRONMENT  1, FROM 514.93 TO 524.93 HOURS,  1 MEN
         PART   1  SQX-3755-420, SURFACE 1, AREA=    .100, TOOL BURDEN=    -0, AREA TOUCHED=-0.     , BURDEN= 4.552E+02
    OPERATION  1, ENVIRONMENT  1, FROM 524.93 TO 534.93 HOURS,  1 MEN
         PART   1  SQX-3755-420, SURFACE 1, AREA=    .100, TOOL BURDEN=    -0, AREA TOUCHED=-0.     , BURDEN= 4.452E+02
```

Table 7. Typical Output, Microbial Burden Prediction Program

B. Description of Subroutines

The following subroutines are described in three groups:

1) Tape alteration program;

2) Burden prediction program;

3) Detailed printout program.

Each group is an independent program and can be run separately if desired, using the first subroutine listed as the main program. All are written in FORTRAN IV and are as machine-independent as possible. Communication between these programs is by means of disk or tape files:

Tape 2: Binary restart dumps (read and write);

Tape 5: Card reader;

Tape 6: Printer;

Tape 9: Binary data written by the tape alteration program for the burden prediction program (read and write),

Tape 11: Binary data saved from burden prediction for detailed printout as desired (read and write);

Tape 12: Binary data originally written by the tape alteration program as Tape 9 but being modified by that program to produce an updated Tape 9. If an older data tape is to be converted to the new format, it is also designated Tape 12. Tape 12 is read only.

Whether these files are on disk or tape depends on whether a tape has been requested or not. If permanent records are desired or if the three programs are to be run separately, tapes should be used since disk files are erased at the end of each job.

1. Tape Alteration Program

SUBROUTINE TAS, the Tape Alteration Subroutine. This subroutine reads tape alteration instruction cards, data cards (if any), and Tape 12 (if used). Tape records are read, altered by subroutine RAS, and checked and written on Tape 9 by subroutine RCWS until the record whose number natches that on the current alteration instruction is encountered. At this time the specified change is made and the next instruction is read. This continues until a higher numbered record is specified in the instruction, at which point the program goes back to reading, checking, and writing the tape records. An instruction to start or stop changing a parameter value does not accomplish this directly but sets appropriate flags in subroutine RAS. See Table 5 for instruction types and format. (See Fig. 1 for TAS macro logic.)

SUBROUTINE CRS, the Card Reading Subroutine. This subroutine reads one card under the appropriate FORMAT statement, except for distribution cards (DQ), of which two or four are read, depending on whether the number of intervals is less or greater than seven.

SUBROUTINE TRS, the Tape Reading Subroutine. This subroutine reads one logical record from Tape 12. This record may contain either 14 or 23 words as described in Section II.A.2.

SUBROUTINE RAS, the Record Alteration Subroutine. This subroutine, by means of arrays specifying for each record type the parameters to be changed and the change to be made, makes all parameter changes specified by any previous tape alteration instruction. These changes continue to be made until another instruction cancels them.

* INSTRUCTION PARAMETERS

N RECORD NUMBER

K CHANGE TYPE

M PARAMETER NUMBER OR NUMBER OF RECORDS

F PARAMETER MODIFIER

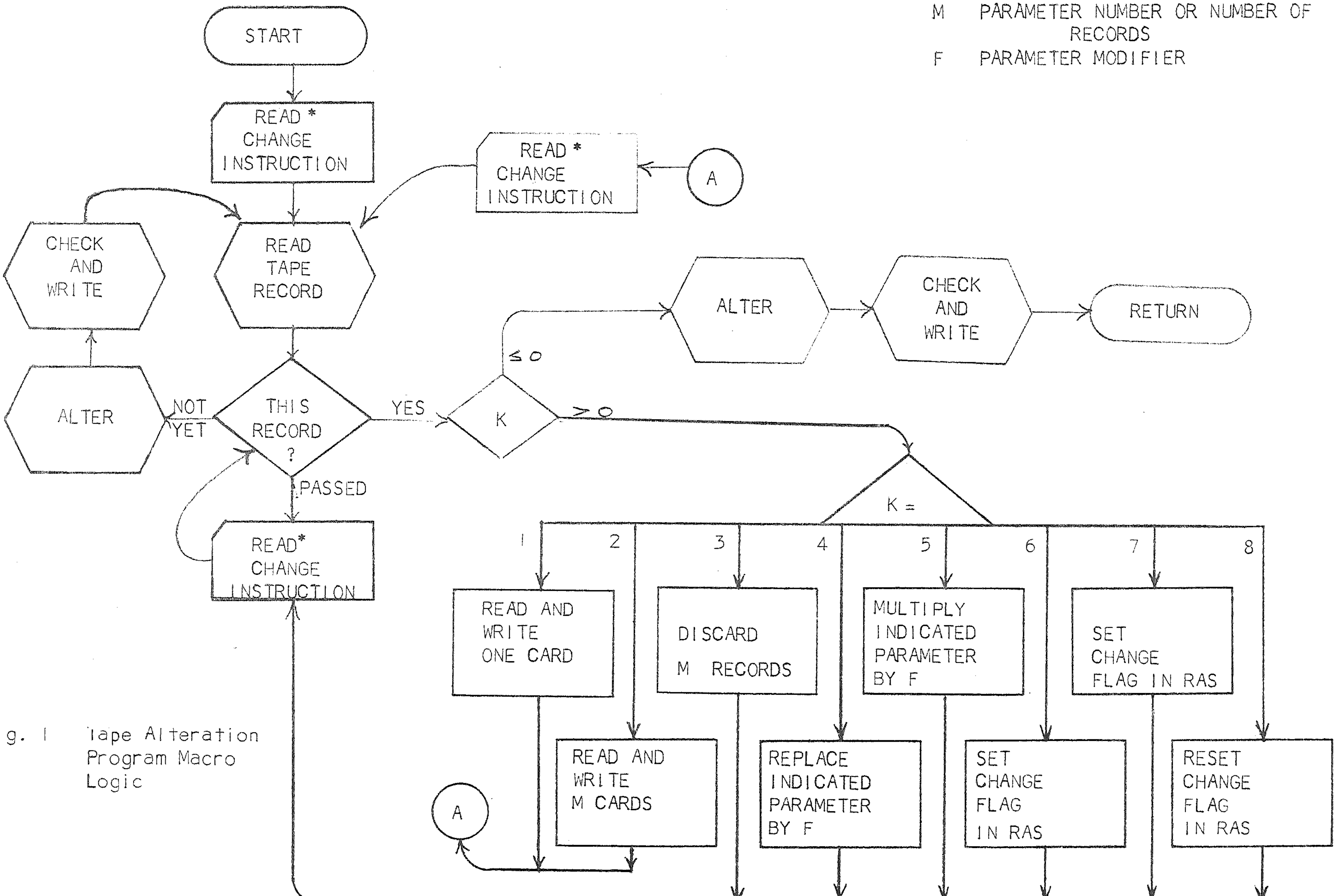

Fig. 1 Tape Alteration Program Macro Logic

SUBROUTINE RCWS, the Record Checking and Writing Subroutine. This subroutine makes all checks that the old Data Check Program used to make (see Table 6 for the error types). The next record type to be read (either from cards or Tape 12) is determined, and the record is then written both on the printer and on Tape 9.

SUBROUTINE EPS, the Error Printing Subroutine. This subroutine prints the appropriate error message and maintains a count of the number of errors of each type. At the end of the run, this subroutine prints the error totals and causes a program exit if any fatal errors occurred.

SUBROUTINE PTS, the Page Titling Subroutine. This subroutine prints parameter numbers along the top of each page for ease in identifying parameters to be changed.

2. Burden Prediction Program

SUBROUTINE BPS, the Burden Prediction Subroutine. This subroutine reads control cards (CC) and performs the actions required by them. Most burden calculations are performed in subroutine MBS, but burden differences and zone reorganization is done in BPS. BPS also performs the restart functions. (See Fig. 2)

SUBROUTINE MBS, the Microbial Buildup Subroutine. This subroutine reads input data for task, subtasks, and operations and computes the microbial burden on each affected surface as a result of fallout, contact or decontamination. See Ref. 2, p 28 for a discussion of this subroutine. (See Fig. 3)

SUBROUTINE HCS, the Histogram Combining Subroutine. This subroutine calculates the probability distribution (in the form of a histogram) of the sum, difference, product, quotient, or maximum value of two histograms. For all but the maximum value, this subroutine requires both input histograms to have their correct mean values in DR(N,1) and will adjust the resulting histogram so that its mean has the correct value. HCS always returns the mean value of the output histogram in

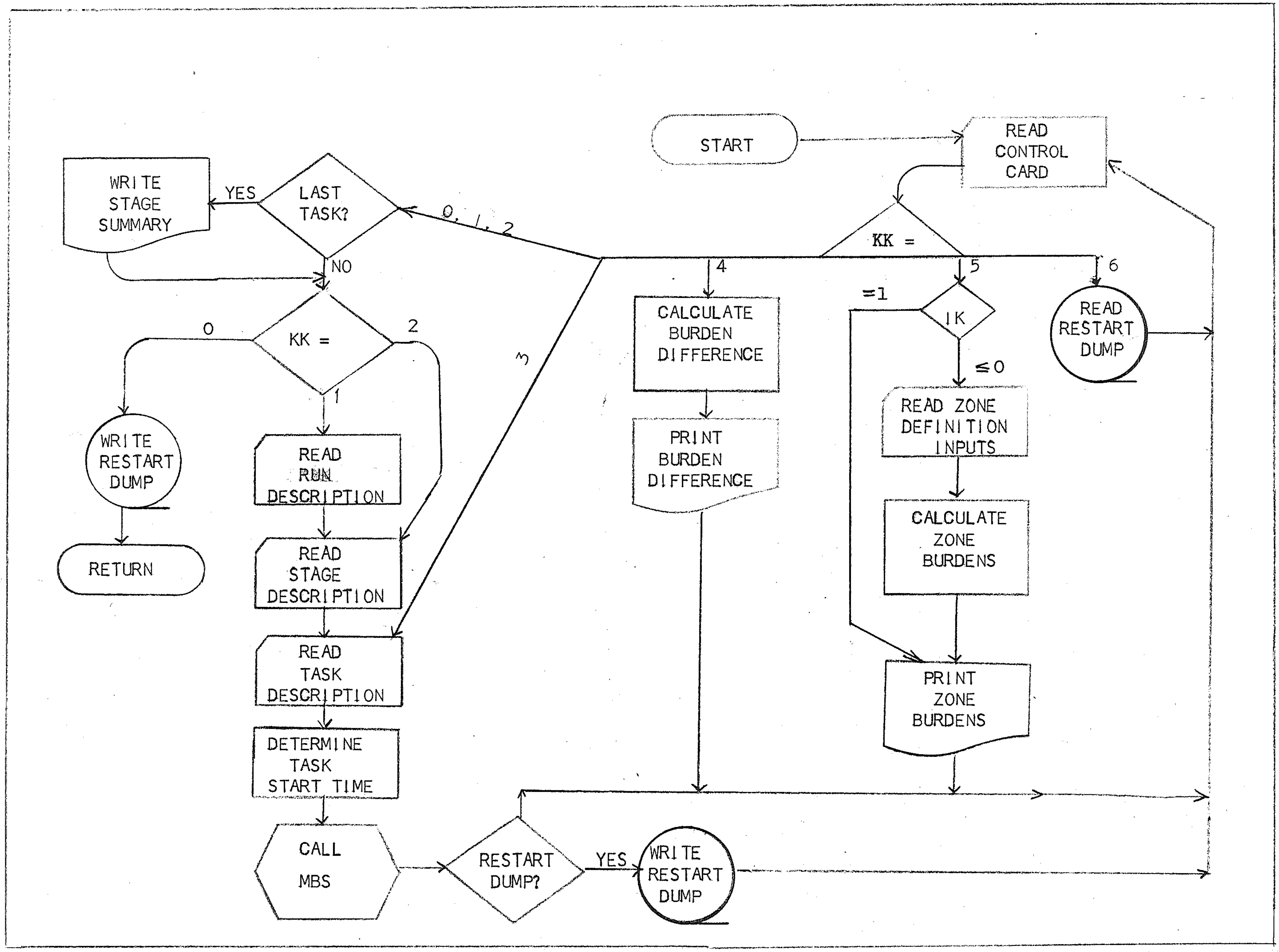

Fig. 2 Burden Prediction Program Macro Logic

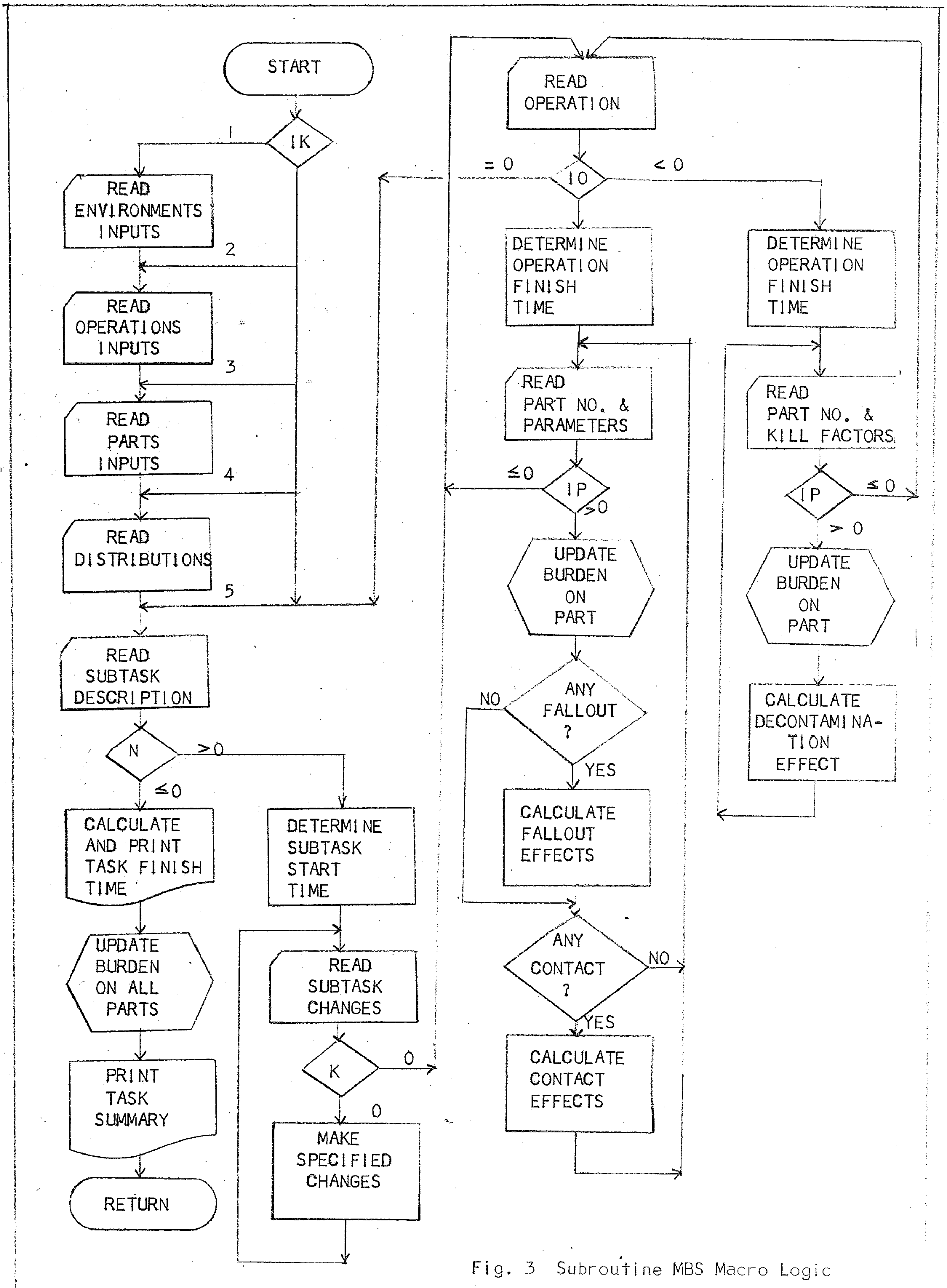

Fig. 3 Subroutine MBS Macro Logic

DR(N,1). See Ref. 2, p 15 for a discussion of this subroutine.

<u>SUBROUTINE HAS</u>, the Histogram Adjusting Subroutine. This subroutine adjusts the number of intervals and the probability levels of a given histogram to match those of the "model" histogram. (See Section II.A.3.)

<u>SUBROUTINE HES</u>, the Histogram Equating Subroutine. This subroutine sets the values of one histogram equal to those of another.

<u>SUBROUTINE HMS</u>, the Histogram Multiplying Subroutine. This subroutine multiplies the abscissae and the mean of a given histogram by a constant.

<u>SUBROUTINE HWS</u>, the Histogram Writing Subroutine. This subroutine prints the mean, probability values, and abscissae of a specified histogram.

<u>SUBROUTINE MAS</u>, the Microbial Accretion Subroutine. This subroutine updates the burden on parts that have not been operated on for some time.

<u>FUNCTION AVF</u>, the Average Value Function. This function computes the mean of a given histogram.

<u>FUNCTION ZF</u>, This function computes the sum, difference, product, quotient, or maximum value of two given numbers (Used by HCS).

3. Detailed Printout Program

<u>SUBROUTINE DPS</u>, the Detailed Printout Subroutine. This subroutine reads detailed printout instruction cards and controls the printing. Since the normal condition is not to print, an instruction is necessary for each task or subtask to be printed. (See Table 2 for detailed printout instructions.)

SUBROUTINE PLS, the Parameter Labeling Subroutine. This subroutine prints the parameters stored on Tape 11 with proper identifying information.

SUBROUTINE HPS, the Histogram Printing Subroutine. This subroutine prints the histogram stored on Tape 11.

4. Subroutines No Longer Used

1) The Data Check Program and its subroutine ECS, the Error Count Subroutine, are no longer used because their function is included in the Tape Alteration Program.

2) Subroutine HNS, the Histogram Numbering Subroutine, was incorporated into subroutine MBS since it was only called twice.

3) The old restart subroutine, MBRS, is not needed because of the new restart capability.

4) The Mean Value Subroutine, MVS, was replaced by the Average Value Function, AVF, to allow greater flexibility in its use.

5) The Mean and Variance Subroutine, MAVS, is not used because AVF now calculates mean values and the variance was not used.

C. INPUT DATA

Table 8 lists all types of input data cards and shows their arrangement. Alternate branches are indicated by the flow lines on the left with the index values that cause the program to take that branch. The brackets on the right indicate cards or groups of cards that are repeated until a blank card occurs. The contents and format for each data card are described below. The data are identified by the variable names used in the computer program (e.g., KK, IK). Their significance is discussed in Reference 2, page 28; and the notation used for these variables in the reference is given in parenthesis in the following list.

Note that each integer constant must be punched next to the right hand side of its field (right adjusted).

CC CONTROL CARD

KK, IK, I1, I2, I3, I4, I5, I6

FORMAT (8I5)

KK=0 causes the program to terminate without reading any additional data.

KK=1 indicates that a RUN is to be initiated and calls for RUN, STAGE, and TASK DESCRIPTION CARDS and TASK data inputs.

KK=2 indicates that a STAGE is to be initiated and calls for STAGE and TASK DESCRIPTION CARDS using TASK data inputs.

TABLE 8

Input Data Deck

	CC	CONTROL CARD
KK=1	RD	RUN DESCRIPTION CARD
KK=2	SD	STAGE DESCRIPTION CARD
KK=3	TD	TASK DESCRIPTION CARD
IK=1	EM	ENVIRONMENTAL SURFACE LIFETIME MODIFIERS
	ED	ENVIRONMENTAL DESCRIPTION CARD
	EQ	ENVIRONMENTAL QUANTITIES CARD
IK=2	OD	OPERATION DESCRIPTION CARD
	OQ	OPERATION QUANTITIES CARD
IK=3	PD	PART DESCRIPTION CARD
	PQ	PART QUANTITIES CARD
IK=4	DD	DISTRIBUTION DESCRIPTION CARD
	DQ	DISTRIBUTION QUANTITIES CARD (1 or 2)
IK=5	KD	SUBTASK DESCRIPTION CARD
	KC	SUBTASK CHANGE CARD
	KØ	SUBTASK OPERATION CARD
	PE	PART EFFECT CARD
KK=5, IK=0	ZD	ZONE DESCRIPTION CARD
	ZC	ZONE COMPOSITION CARD

KK=3 indicates that a TASK is to be initiated and calls for a TASK DESCRIPTION CARD and TASK data inputs.

For KK=1, 2, 3, IK determines the point at which TASK inputs start. These inputs and the values of IK for which they are read are:

IK=1	ENVIRONMENTS (EM and all subsequent data)
IK=2	OPERATIONS (OD and all subsequent data)
IK=3	PARTS (PD and all subsequent data)
IK=4	DISTRIBUTIONS (DD and all subsequent data)
IK=5	SUBTASKS (KD and all subsequent data)

I1, I2, I5, and I6 are used as follows (I3 and I4 are not used)

I1 controls the level of detail in printout:

= 0	No change
= 1	Print subtask information
= 2	Start operation level printout
= 3	Start part level printout

I2 controls the restart card dumps:

= 0	No change
= 1	Restart dump at end of run only
= 2	Restart dump at end of each task
= 3	No restart dumps.

I5, if non-zero, is the number of the distribution to be used for adjusting the probability levels (See Ref. 4, p 50-51).

I6, if non-zero, is the new task environment number. I6 must be specified if any fallout is to be determined, but it need be specified only on the first control card.

KK=4 calls for determining and printing the burden difference between the burdens stored in special storage distributions I1 and I2. (These distributions have a numbering sequence separate from the distributions specified by DD cards.) Storage in these distributions is accomplished by specifying the distribution number (1 through 20) on the TASK DESCRIPTION CARD (TD).

KK=5 calls for the printing of areas and microbial burdens by zones.

IK=0 indicates that the zones are to be defined by a deck of ZONE DEFINITION CARDS.

IK=1 indicates that zones are the same as parts and that no ZONE DEFINITION CARDS are necessary. This card is to be followed by another CONTROL CARD.

KK=6 calls for a restart and requires a previously written restart dump tape to be loaded on Tape 2.

A restart contol card should be the first card in the data (card or tape) since any previous data is incorporated in the restart dump. When encountered by the burden prediction program, a restart control card will cause a restart tape to be read.

RD RUN DESCRIPTION CARD

KR, RUN

FORMAT (I5, 1X, 7A6)

A RUN DESCRIPTION CARD is the first card to follow the CONTROL CARD initiating a run. KR is the run number and RUN is a 42 alphameric character description of the run.

SD STAGE DESCRIPTION CARD

KS, STG

FORMAT (I5, 1X, 7A6)

A STAGE DESCRIPTION CARD follows a RUN DESCRIPTION CARD or a CONTROL CARD initiating a Stage. KS, which must be in the range 1 to 20 is the stage number and STG is a 42 alphameric character description of the Stage.

TD TASK DESCRIPTION CARD

KT, TSK, L1, L2, L3, L4

FORMAT (I5, 1X, 7A6, 2X, 4I5)

A TASK DESCRIPTION CARD follows a STAGE DESCRIPTION CARD or a CONTROL CARD initiating a task. KT, which must be in the range 1 to 100, is the task number and TSK a 42 alphameric character description of the task. L1 and L2 indicate the finish time special storage distributions of prerequisite tasks which are to determine the start time of this task. If only one prerequisite (L1) is specified, the new start time is set equal to the time stored in L1. If no prerequisite is specified, the start time is set equal to zero. L3 is the index of the finish time

special storage distribution if this task is to be prerequisite for any later task. L4, if specified, causes the total microbial burden at the end of the task to be stored in special storage distribution L4 and to be available for later computation of a burden difference. IK on the preceding CONTROL CARD indicates the card type to follow. L1, L2, L3, and L4 are in the range 1 to 20.

EM ENVIRONMENTAL AVERAGE LIFETIME MODIFIERS CARD

AES(1), AES(2), AES(3), AES(4)

FORMAT (4E10.3)

The ENVIRONMENTAL AVERAGE LIFETIME MODIFIERS appear on the first card of the environments inputs. AES(J) is the factor which modifies the quantity AET(J) on card EQ.

ED ENVIRONMENTAL DESCRIPTION CARD

N, DSC

FORMAT (I5, 1X, 4A6)

Each of the (up to 10) environments requires two input cards. The first of these is the ENVIRONMENT DESCRIPTION CARD which gives the index N and the 24 alphmeric character description DSC. The last such description card is to be blank to indicate that all environment inputs have been read. All but the last (blank) ENVIRONMENT DESCRIPTION CARD are to be followed by an ENVIRONMENT QUANTITIES CARD.

EQ ENVIRONMENT QUANTITIES CARD

IEC(N), IET(N), AEC(N), AET(N), AED(N), AEF(N,1), AEF(N,2), AEF(N,3), AEF(N,4)

FORMAT (2I5, 7E10.3)

IEC(N),AEC(N) describe the background number of organisms/cu ft (C) in environment N. IEC is a "shape" distribution index and AEC is the mean of the distribution for this application.

IET(N), AET(N) are the average lifetime distribution shape and mean (V).

AED(N) is the mean number of organisms per cubic foot of air per man in this environment (d).

AEF(N) is the fallout velocity mean value (f).

OD OPERATION DESCRIPTION CARD

N, DSC

FORMAT (I5, 1X, 4A6)

Each of the (up to 20) cataloged operations requires two input cards. The first of these is the OPERATION DESCRIPTION CARD which gives the index N and the 24 alphameric character description DSC. The last of these cards is to be blank to indicate that all operations inputs have been read. All but the last (blank) card are to be followed by an OPERATION QUANTITIES CARD.

OQ: OPERATION QUANTITIES CARD

IOT(N), IOQ(N), IOC(N)

FORMAT (3I5)

IOT(N) is the distribution shape for the time required to perform operation N.

IOQ(N) is the distribution shape for the number of organisms per cubic foot per man.

IOC(N) is the distribution shape for the number or organisms on hand or tool (contact contamination).

PD: PART DESCRIPTION CARD

N, DSC

FORMAT (I5, IX, 6A6)

Each of the (up to 120) initial parts requires two input cards. The first is the PART DESCRIPTION CARD giving the part number N and the 36 alphameric character description DSC. The last 12 characters (in columns 31 through 42) are retained as a permanent serial description of the part. The last of these cards is to be blank to indicate that there are no more parts inputs. All but the blank card are followed by a PART QUANTITIES CARD.

PQ: PART QUANTITIES CARD

IAB(N,1), IAB(N,2), IAB(N,3), IAB(N,4), IAB(N,5), IAB(N,6).

AAG(N), AAS(N)

FORMAT (6I5, 2E10.3)

IAB(1) is the index of the distribution of microbial burden on the top surface of the part. The area of this and the other surfaces is given in the coefficient associated with the distribution (see the explanation for DQ below).

IAB(2) is the index of the distribution for area and burden of all other exterior surfaces.

IAB(3) is the index of the distribution for area and burden of the mated surface.

IAB(4) is the index of the distribution for area and burden of the occluded surface.

IAB(5) is the environment of the part used in updating microbial accretion for parts that have not been operated on for some time.

IAB(6) is the distribution shape for part contact retention (S1).

AAG is the surface retention factor for fallout contamination (g).

AAS is the mean part contact retention (S1).

DD: DISTRIBUTION DESCRIPTION CARD

N, DSC, L, M

FORMAT (I5, 1X, 4A6, 2I5)

Arrays are defined for storing up 500 histograms, each with a maximum of 11 absisssa values and their associated probabilities. Since the program automatically indexes new histograms that it must store, only a fraction of these 500 should be specified at input time. The DISTRIBUTION DESCRIPTION CARD gives the distribution number N and 24 alphameric character description DSC. The last of these cards is to be blank to indicate that there are no more distributions inputs.

The index L indicates whether the range values and associated probabilities are to be read directly from the DISTRIBUTION QUANTITIES CARDS or calculated from the range minimum, mode and maximum values.

M is the number of values (to be in the range 1 to 11) If M=1, the distribution is a constant with value XR(N,1).

DQ: DISTRIBUTION QUANTITIES CARDS

For L = 1: DR(N,J), J=1,M

XR(N,J),J=1,M

For L = 2: DR(N,1),X1,X2,X3

FORMAT (8E10.3)

For L = 1, there are two sets of data DR(N,J) and XR(N,J). DR(N,1) is equal to the area for distributions describing the microbial burden on particular surfaces. For other distributions it is left blank and the program automatically supplies the mean of the distribution. DR(2) to DR(M) are the probabilities associated with the intervals XR(1) to XR(2), XR(2) to XR(3),...XR(M-1) to XR(M). If M is larger than 8, two cards are required for each set of array values.

For L = 2, DR(1) has the same significance as for L=1. The range intervals of the distribution and the associate probabilies are determined by the program to describe, with a 5 value histogram, a triangular distribution with the range X1 to X3 and mode (probability peak) at X2.

The program automatically generates and indexes (index = KO), a zero value histogram.

KD: SUBTASK DESCRIPTION CARD

N, DSC, N1, N2

FORMAT (I5, 1X, 4A6, 2I5)

The input data for each of the (up to 20) subtasks per task is headed with a SUBTASK DESCRIPTION CARD that gives the subtasks number N, the 24 alphameric character description DSC, and the indices of two prerequisite subtasks, N1 and N2. If N1 and N2 are blank, the subtask start time is taken to be the same as the task start time. If there is only one prerequisite subtasks, it is to be indicated by N1 since, if N1 is blank, N2 is not checked. Both subtasks must have been previously considered by the program and had their finish times computed.

KC: SUBTASK CHANGE CARD

K, IR(1), IR(2), IR(3), IR(4), IR(5), AR(1), AR(2), AR(3)

FORMAT (6I5, 3E10.3)

The SUBTASK CHANGE CARD is used to define any changes in area, environment, or retention factor values for a part. The amount of area AR(1), is subtracted from part IR(3), surface IR(4), and added to part IR(1), surface IR(2).

If IR(5), is a positive integer, it is the new environment number assigned to part IR(1).

If AR(2) is a non-zero number, it is the new surface retention factor (g) for fallout for part IR(1).

If AR(3) is a non-zero number, it is the new mean part contact retention factor (S1) for contact for part IR(1).

K is the change index and is to be an arbitrary positive integer except where it is to indicate there are no more changes, in which case it is left blank.

KO: SUBTASK OPERATION CARD

IO, IKE AKT, AKQ

FORMAT (2I5, 2E10.3)

Each subtask consists of an arbitrary number of operations. These operations are partially described in the catalog of operations (see inputs OD and OQ described above) but have additional parameters to describe the resulting microbial burden changes for each particular performance of the operation.

IO is the operation number. If IO is zero or blank, there are no more operations and the program proceeds to a KD card. If IO is negative, a decontamination is indicated for which the input quantities have a a different significance.

For decontamination ($IO<0$):

IKE is not significant and can be left blank.

AKT is the operation time.

For fallout and contact ($IO>0$):

IKE is the operation environment unless it is left blank, in which case the operation environment is the same as for the previous operation. If IKE is blank for all operations in a subtask, the operation environments are all the same as the environment number I6 given in card CC.

AKT is the mean operation time and is used to modify the distribution IOT(IO) (see card OQ) to determine the operation time distribution.

AKQ is the number of men close to the hardware during this operation.

PE: PART EFFECT CARD

IP, LS, APD, APC, APS, APA(1), APA(2) or IP, LK, AR(1), AR(2), AR(3), AR(4)

FORMAT (2I5, 5E10.3)

IP is the number of the part affected. If IP is blank or zero, it indicates that all affected parts have been considered and the program is to proceed to a consideration of the next operation.

For fallout and contact operations:

LS is the tool retention distribution for contact.

APD is not used.

APC is the mean number of organisms on hand or tool (contact contamination).

APS is the retention factor for the tool in determining contamination by contact (S2).

APA(1) is the contact area for the top surface, APA(2) the contact area for other exterior surfaces. It is assumed that occluded and mated surfaces are not contacted.

For decontamination operations:

LK is the distribution shape of the fraction of organisms removed.

AR(J) is the mean fraction of organisms removed from surface J.

ZD: ZONE DESCRIPTION CARD

IZ, DSC

FORMAT (I5, 1X, 4A6)

Defintion of each thermal zone in terms of the fractions of surfaces that comprise it requires a ZONE DEFINITION CARD for the zone and one ZONE COMPOSITION CARD for each part involved. The zone is described in terms of a number, IZ, and a 24 alphameric character description DSC. If IZ is blank or zero it indicates that all zones have been defined and that the program is to proceed to the next CONTROL CARD.

ZC: ZONE COMPOSITION CARD

IP, FP(1), FP(2), FP(3), FP(4)

FORMAT (I5, 5X, 4F10.7)

A separation ZONE COMPOSITION CARD is used for each part IP that is wholly or partially included in zone IZ. The quantities FP(J) indicate the decimal fractions of surface J (top, exterior, mated, and interior) to be included in zone IZ. The last ZONE COMPOSITION CARD for each zone must be blank; this is followed by the next ZD card.

D. JOB CONTROL AND DECK ARRANGEMENT

Table 9 lists a typical input deck for the Microbial Burden Prediction Program. the control cards are for the Univac 1108 computer.

It is assumed that a previous run has written a data tape (6524) and that it is desired to use this tape, add some more data from cards, and compute a burden prediction. Detailed information from the burden calculations is to be written on Tape 6523, and is to be printed for the last task computed (Stage 5, Task 7). A new data tape (6522) is also to be written.

Tape 6522 is designated Tape 9 and will replace 6524, which is designated Tape 12. (Tape 6524 was designated Tape 9 when it was written during the previous run.) Tape 6523 is designated Tape 11, and will contain detailed information that can be selected for printing at some later date.

The main program contains only four call statements; it calls each of the major subprograms (Tape Alteration, Burden Prediction, and Detailed Printout) in turn and exits. The remainder of the source deck is as listed in the Appendix.

The first data card is read by the Tape Alteration Subroutine (TAS). It indicates that parameter number 3 on record number 103 is to be changed to a 7. The result of course depends on what type of card record 103 represents; it could, for example, be a change in the number of histogram intervals to be used in the computations. The second data card is also read by TAS; it causes (up to) 999 cards to be read in after record number 1755. These records and the records obtained from Tape 12 are written on Tape 9 to form an updated data tape. Note that it is not necessary to count the cards to be added as long as they are added to the end of the deck. Any number

TABLE 9. ARRANGEMENT OF DATA DECK

```
ΔRUN,/TPC  LBB,06024S,07695,05,199/000   . MBP    L BUSCH  125/43
ΔMSG MOUNT TAPE 6522 TO WRITE LBB
ΔMSG MOUNT TAPE 6523 TO WRITE LBB
ΔASG,T 9,T,6522W
ΔASG,T 11,T,6523W
ΔASG,T 12,T,6524R
ΔFOR
      CALL TAS
      CALL BPS
      CALL DPS
      CALL EXIT
      END
ΔFOR
      SUBROUTINE TAS

  ----- REMAINDER OF SOURCE DECK ----

      END
ΔXQT
       103    4    3    7                                 (TAS INSTRUCTION)
      1755    2  999                                      (TAS INSTRUCTION)
    2    1                       6                        (CONTROL CARD)
    7 S/C ASSEMBLY + TEST, PASADENA
    5 INSTALL BAY VII SOLAR PANEL                                    11    2
    1.         1.       1000.        5.
    5 HIGH BAY
    6    6     1.          1.          1.       100.       100.       100.
    0
    0
    5 SOLAR PANEL, BAY VII     SQX-3762-599
   25                              1     1.          1.
    0
    6 ONE BUMP ID HISTOGRAM           1     2
               1.
    1.         2.
    5 HIGH BAY SURFACE DISTR          1     2
               1.
     .99       1.
   25 SOLAR PANEL, EXTERNAL           1     2
    1.         1.
     .99       1.
    0
    5 INSP OF SOLAR PANEL

  --- REMAINDER OF DATA DECK ---

    0                                                     (CONTROL CARD)
    1    5    7                                           (DPS INSTRUCTION)
    0                                                     (DPS INSTRUCTION)
ΔPMD,AE
ΔFIN
```

--- NOTE THAT Δ REPRESENTS A MASTER SPACE (MULTIPLE 7-8 PUNCH).

larger than the number of cards to be read can be used for the parameter M on the TAS instruction; reading of these cards will be terminated after the last control card since this control card indicates the end of the data deck. (By inference, record number 1756 on Tape 12 was a control card indicating the end of the previous data deck. This record does not appear on Tape 9, being replaced in the record sequence with a different control card.)

Immediately after the last control card (the last card read by TAS) are two instructions to the Detailed Printout Subroutine (DPS). The first of these indicates that all detailed information resulting from burden calculations on Task 7 of Stage 5 are to be printed. The last DPS instruction amounts to a program stop; its omission would cause an abnormal exit since DPS reads another instruction as soon as it has completed executing one.

III. RESULTS AND CONCLUSIONS

In general, all objectives of Phase VII were met; although the results in some cases differed from what was expected.

The restart capability performed satisfactorily; but since no restart information is provided for the tape alteration program, a number of false error messages (e.g., "prerequisite missing") are generated during data checks. These error messages can of course be ignored but may possibly obscure a valid error message. A restart capability could be added to the tape alteration program, but it would require an extra tape and was therefore decided against.

The detailed printout capability performs well and is easy to use; it promises to be a very useful addition to the program. The ability to dispense with tapes for short runs and to read all inputs from cards is a great convenience.

The tape alteration program does everything that was expected of it. This capability should be particularly useful for sensitivity studies.

Although no extensive comparisons were made, it appears that the running time of the burden prediction model does vary as the square of the number of histogram intervals used. Calculation of the Mariner Venus 67-2 assembly and test sequence (using data prepared in Phase VI) required 20 minutes using 5-interval histograms and seven minutes using three intervals.

The re-formatted outputs were somewhat disappointing since the print-out volume was reduced only by approximately 20%. This is because the addition of a complete task summary (burden histograms for each surface of each part) almost offset the reduction obtained by re-formatting. It was concluded that such detailed task summaries should be made optional.

No trouble was experienced in getting the computer program to run on the Univac 1108 at JPL. (It was originally checked out on the CDC 6400 at Martin Marietta's Denver facility.) However, some modification of the program was necessary to make effective use of the JPL computer. In particular, the restart dumps were changed from punched cards to tape.

Although the purpose of Phase VII was not to produce a burden prediction, a prediction for the Mariner Venus 67-2 spacecraft was generated by the demonstration case (Phase VI data). Comparison of the Phase VII prediction mean value with that obtained in Phase VI was made at two points during the runs:

	Phase VI	Phase VII
Stage 1, Task 81 (End of Pasadena Operations)	121600.	202000.
Stage 2, Task 15 (Burden at launch)	79700.	207000.

Most of this difference is attributed to an error that was discovered in the burden prediction model during Phase VII (a reference histogram was being changed inadvertently). Part of the difference is due to the higher contact burden (corresponding to vegetative organisms) use in Phase VII; it was not judged worth the expense to make a special computer run in Phase VII merely to alter this parameter. The Phase VII burden prediction was compared with two estimates obtained by JPL as described in Ref. 6:

Long-term cupons (Ref. 6, Table 4, May 28): 3.2×10^5 spores

Swab-rinse (Ref. 6, Table 6, May 28): 3.9×10^5 spores

The Phase VII calculation showed that the burden will exceed these values with probability 0.17 and 0.10, respectively.

IV. RECOMMENDATIONS FOR PROGRAM CONTINUATION

The work performed during Phase VII was primarily intended to improve the flexibility of the Microbial Burden Prediction Model so that sensitivity studies and parameter adjustments could be more easily made. Although the Mariner Venus 67-2 data prepared in Phase VI could be used for this purpose, the Mariner Mars 69 data are now available and incorporate a number of improved methods for bioassay and assembly sequence recording. It is therefore recommended that a set of inputs be prepared to simulate the Mariner Mars 69 assembly and test sequence so that sensitivity and parameter adjustment studies can be performed on the best available data.

Since the routine manual effort of card punching, sorting, and collating formed a considerable part of the Phase VI work, it is also recommended that a computer program be written to perform the routine work associated with input preparation. Such a program should also include a complete list of parts at the level of detail recorded in the assembly log, and should be able to maintain a running account of the status (i.e., installed or not) of each part. This computer program should be prepared with the capability of adaptation to a real time burden prediction capability for use on future spacecraft assemblies.

An additional recommendation is to make the bulk of the Task Summary printout optional; that is, omit printing the zone-by-zone burdens unless requested. It may be preferable to permit this option at the subtask level as well.

REFERENCES

The following references together form the final report for JPL Contract 952028, A Study Program on the Development of a Mathematical Model(s) for Microbial Burden Prediction:

1) Volume I, Technical Report
2) Volume II, User's Manual for the Microbial Burden Prediction Model
3) Volume III, Appendices
4) Volume IV, Addendum: Technical Report
5) Volume V, Addendum: Appendices

All were issued in December, 1968 and are available through the NASA STAR.

6) Christensen, M.R., Green, R. H, and Stern, J. A. Microbiological Monitoring of the Mariner V Spacecraft, 604-54, Jet Propulsion Laboratory, Pasadena, California, July 10, 1969.

APPENDIX

PROGRAM SOURCE DECK LISTINGS

```
      SUBROUTINE TAS
C     TAPE ALTERATION SUBROUTINE
C     THIS SUBROUTINE READS FROM TAPE12 AND WRITES ON TAPE9
      COMMON/T/LN,LR,IX(6),XX(22),M,N1,N2,I(6,20),X(22,20),K(22,20),
     .KIL,NE,KE(11),IAB(120,6),DR(500),XR(500),ND,IO,AR(4),KKT(20),NP,
     .LH(20),KXT                                                          A
  600 FORMAT(/13H ----- CHANGE,I10,2I5,F15.6/)
      KXT=0                                                               A
      DO 1000 L=1,11
 1000 KE(L)=0
      N1=0
      N2=0
      KIL=1
      ND=0
      NE=0
      LN=1
      CALL PTS
      DO 1 J=1,22
      DO 1 L=1,20
    1 K(J,L)=0
      READ(5,500) NN,KK,MM,F,KLL
      IF((NN.NE.0).OR.(KK.NE.2)) GO TO 9
      NP=NP+3
      IF(NP.GE.55) CALL PTS
      WRITE(6,600) NN,KK,MM,F
      GO TO 21
    2 READ(5,500) NN,KK,MM,F,KLL
  500 FORMAT(I10,2I5,F10.0,I5)
    9 IF(KLL.GT.0) KIL=KLL
    3 CALL TRS
    4 IF(NN-N1) 6666,7,5
    5 CALL RAS
      CALL RCWS
      IF(LN.EQ.0) RETURN
      GO TO 3
 6666 NP=NP+5
      IF(NP.GE.55) CALL PTS
      WRITE(6,601)
  601 FORMAT(/34H ***** INSTRUCTION OUT OF SEQUENCE)
      WRITE(6,600) NN,KK,MM,F
    6 READ(5,500) NN,KK,MM,F,KLL
      IF(KLL.GT.0) KIL=KLL
      GO TO 4
    7 NP=NP+3
      IF(NP.GE.55) CALL PTS
      WRITE(6,600) NN,KK,MM,F
      IF(KK.GT.0) GO TO 8
      CALL RAS
      CALL RCWS
      CALL EPS(0)
      RETURN
    8 GO TO(101,102,103,104,105,106,107,108),KK
C     REPLACE RECORD NN WITH A NEW CARD
  101 CALL CRS
      CALL RCWS
      IF(LN.EQ.0) RETURN
      GO TO 2
C     ADD MM CARDS AFTER RECORD NN
  102 CALL RAS
      CALL RCWS
```

```
      IF(LN.EQ.0) RETURN
   21 DO 20 L=1,MM
      CALL CRS
      CALL RCWS
      IF(LN.EQ.0) RETURN
   20 CONTINUE
      GO TO 2
C     DELETE MM RECORDS STARTING WITH RECORD NN
  103 DO 30 L=1,MM
      CALL TRS
   30 CONTINUE
      GO TO 6
C     REPLACE PARAMETER MM ON RECORD NN BY F
  104 IF(LR.EQ.17) GO TO 40
      IF(MM.LE.6) GO TO 41
      XX(MM-6)=F
      GO TO 6
   41 IX(MM)=F
      GO TO 6
   40 XX(MM)=F
      GO TO 6
C     MULTIPLY PARAMETER MM ON RECORD NN BY F
  105 IF(LR.EQ.17) GO TO 50
      IF(MM.LE.6) GO TO 6
      XX(MM-6)=XX(MM-6)*F
      GO TO 6
   50 XX(MM)=XX(MM)*F
      GO TO 6
C     REPLACE PARAMETER MM BY F ON NN AND SUBSEQUENT RECORDS
  106 K(MM,LR)=1
   60 IF(LR.EQ.17) GO TO 66
      IF(MM.LE.6) GO TO 64
      X(MM-6,LR)=F
      GO TO 6
   64 I(MM,LR)=F
      GO TO 6
   66 X(MM,LR)=F
      GO TO 6
C     MULTIPLY PARAMETER MM BY F ON NN AND SUBSEQUENT RECORDS
  107 K(MM,LR)=2
      GO TO 60
C     STOP ALTERING PARAMETER MM.  (RECORD NN IS NOT ALTERED.)
  108 K(MM,LR)=0
      GO TO 6
      END
```

```
      SUBROUTINE TRS
C     TAPE READING SUBROUTINE  (READS BINARY RECORDS FROM TAPE12)
      COMMON/T/LN,LR,IX(6),XX(22),M,N1,N2,I(6,20),X(22,20),K(22,20),
     .KIL,NE,KE(11),IAB(120,6),DR(500),XR(500),ND,IO,AR(4),KKT(20),NP,
     .LH(20),KXT                                                        A
      N1=N1+1
      IF(LN.EQ.17) GO TO 5
      READ(12)LR,IX,(XX(J),J=1,7)
      RETURN
    5 READ(12)LR,XX
      RETURN
      END
```

```
      SUBROUTINE CRS
C     CARD-READING SUBROUTINE
      COMMON/T/LN,LR,IX(6),XX(22),M,N1,N2,I(6,20),X(22,20),K(22,20),
     .KIL,NE,KE(11),IAB(120,6),DR(500),XR(500),ND,IO,AR(4),KKT(20),NP,
     .LH(20),KXT                                                        A
      LR=LN
      GO TO(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,5),LN
    1 READ(5,501) IX
  501 FORMAT(4I5,10X,2I5)
      RETURN
    2 READ(5,502) IX(1),(XX(J),J=1,4),IX(2),IX(3)
  502 FORMAT(I5,1X,4A6,2I5)
      RETURN
    3 READ(5,503) IX,(XX(J),J=1,3)
  503 FORMAT(6I5,3E10.3)
      RETURN
    4 READ(5,505) IX(1),IX(2),XX(1),XX(2)
      RETURN
    5 READ(5,505) IX(1),IX(2),(XX(J),J=1,5)
  505 FORMAT(2I5,7E10.3)
      RETURN
    6 READ(5,508) IX(1),(XX(J),J=1,7)
      RETURN
    7 READ(5,508) IX(1),(XX(J),J=1,7)
      RETURN
    8 READ(5,508) IX(1),(XX(J),J=1,7),(IX(J),J=2,5)
  508 FORMAT(I5,1X,7A6,2X,4I5)
      RETURN
    9 READ(5,509) (XX(J),J=1,4)
  509 FORMAT(8E10.3)
      RETURN
   10 READ(5,502) IX(1),(XX(J),J=1,4)
      RETURN
   11 READ(5,505) IX(1),IX(2),(XX(J),J=1,7)
      RETURN
   12 READ(5,502) IX(1),(XX(J),J=1,4)
      RETURN
   13 READ(5,501) (IX(J),J=1,3)
      RETURN
   14 READ(5,508) IX(1),(XX(J),J=1,6)
      RETURN
   15 READ(5,503) IX,XX(1),XX(2)
      RETURN
   16 READ(5,502) IX(1),(XX(J),J=1,4),IX(3),IX(2)
      RETURN
   17 READ(5,509) (XX(J),J=1,M)
      MP=M+11
      READ(5,509) (XX(J),J=12,MP)
      RETURN
   18 READ(5,502) IX(1),(XX(J),J=1,4)
      RETURN
   19 READ(5,519) IX(1),(XX(J),J=1,4)
  519 FORMAT(I5,5X,4E10.7)
      RETURN
      END
```

```
      SUBROUTINE RAS
C     RECORD ALTERATION SUBROUTINE
      COMMON/T/LN,LR,IX(6),XX(22),M,N1,N2,I(6,20),X(22,20),K(22,20),
     .KIL,NE,KE(11),IAB(120,6),DR(500),XR(500),ND,IO,AR(4),KKT(20),NP,
     .LH(20),KXT
      IF(LR.EQ.17) GO TO 29
      DO 10 L=1,6
      IF(K(L,LR).EQ.1) IX(L)=I(L,LR)
   10 CONTINUE
      DO 20 L=7,13
      IF(K(L,LR)-1) 20,21,22
   21 XX(L-6)=X(L-6,LR)
      GO TO 20
   22 XX(L-6)=XX(L-6)*X(L-6,LR)
   20 CONTINUE
      RETURN
   29 DO 30 L=1,22
      IF(K(L,LR)-1) 30,31,32
   31 XX(L)=X(L,LR)
      GO TO 30
   32 XX(L)=XX(L)*X(L,LR)
   30 CONTINUE
      RETURN
      END
```

A

```
      SUBROUTINE RCWS
C     RECORD CHECKING AND WRITING SUBROUTINE  (WRITES BINARY ON TAPE9)
C     LR IS CURRENT RECORD TYPE, LN IS NEXT RECORD TYPE.
      COMMON/T/LN,LR,IX(6),XX(22),M,N1,N2,I(6,20),X(22,20),K(22,20),
     .KIL,NE,KE(11),IAB(120,6),DR(500),XR(500),ND,IO,AR(4),KKT(20),NP,
     .LH(20),KXT                                                        A
      DIMENSION CT(20)
      DATA(CT(I),I=1,20)/2HCC,2HKD,2HKC,2HKO,2HPE,2HRD,2HSD,2HTD,2HEM,
     .2HED,2HEQ,2HOD,2HOQ,2HPD,2HPQ,2HDD,2HDQ,2HZD,2HZC,2HPE/
  600 FORMAT(I7,3X,A2,I4,1H/,I3,1H/,I2)
      N2=N2+1
      LX=LN
      NP=NP+1
      IF(LR.EQ.17) NP=NP+2
      IF(NP.GE.55) CALL PTS
      GO TO(10,20,30,40,50,60,70,80,90,100,110,120,130,140,150,160,170,
     .180,190,200),LR
C                                   CC CARD
   10 WRITE(6,600) N2,CT(LR)
      WRITE(6,601) IX
  601 FORMAT(1H+,23X,6I4)
      KK=IX(1)
      IF(KK.GT.0) GO TO 11
      WRITE( 9) LR,IX,(XX(J),J=1,7)
      CALL EPS(0)
      LN=0
      RETURN
   11 IF(KK.LE.5) GO TO 12
      KXT=1                                                             A
      LN=1                                                              A
      IF(KK.EQ.6) GO TO 1000                                            A
      KXT=0                                                             A
      CALL EPS(1)
      LN=8
      GO TO 1000
   12 IF(KK-4) 13,14,15
   13 IK=IX(2)
      LN=KK+5
      IF((IK.GT.0).AND.(IK.LE.5)) GO TO 1000
      CALL EPS(1)
      IK=5
      GO TO 1000
   14 IF((IX(3).LE.0).OR.(IX(3).GT.20)) CALL EPS(1)
      IF((IX(4).LE.0).OR.(IX(4).GT.20)) CALL EPS(1)
      LN=1
      GO TO 1000
   15 LN=18
      IF(IX(2).EQ.1) LN=1
      GO TO 1000
C                                   KD CARD
   20 N=IX(1)
      WRITE(6,600) N2,CT(LR),KS,KT,N
      WRITE(6,602) (IX(J),J=1,3),(XX(J),J=1,4)
  602 FORMAT(1H+,23X,3I4,17X,4A6)
      IF(N.GT.0) GO TO 21
      LN=1
      GO TO 1000
   21 LN=3
      IF(N.GT.20) GO TO 28
      IF(KKT(N).EQ.1) CALL EPS(5)
      IF(IX(2).LE.0) GO TO 29
```

```
      M1=IX(2)
      IF(M1.GT.20) GO TO 28
      IF(KKT(M1).EQ.0) CALL EPS(2)
      IF(IX(3).LE.0) GO TO 29
      M2=IX(3)
      IF(M2.GT.20) GO TO 28
      IF(KKT(M2).EQ.0) CALL EPS(2)
      GO TO 29
   28 CALL EPS(1)
   29 KKT(N)=1
      GO TO 1000
C                                     KC CARD
   30 WRITE(6,600) N2,CT(LR),KS,KT,N
      WRITE(6,603) IX,(XX(J),J=1,3)
  603 FORMAT(1H+,23X,6I4,3F10.3)
      IF(IX(1).GT.0) GO TO 32
      LN=4
      GO TO 1000
   32 IF(XX(1).GT.0.) GO TO 33
      IF(IX(2)) 38,38,39
   33 I1=IX(2)
      I2=IX(3)
      IF((I1.LE.0).OR.(I1.GT.120)) GO TO 38
      IF((I2.LE.0).OR.(I2.GT.4)) GO TO 38
      IB=IAB(I1,I2)
      I3=IX(4)
      I4=IX(5)
      IF((I3.LE.0).OR.(I3.GT.120)) GO TO 38
      IF((I4.LE.0).OR.(I4.GT.4)) GO TO 38
      IA=IAB(I3,I4)
      IF(IA.GT.0) GO TO 34
      CALL EPS(3)
      GO TO 39
   34 IF(IB.GT.0) GO TO 35
      ND=ND+1
      IB=ND
      IAB(I1,I2)=IB
      DR(IB)=0.
   35 DR(IA)=DR(IA)-XX(1)
      DR(IB)=DR(IB)+XX(1)
      WRITE(6,633) DR(IB),DR(IA)                                              A
  633 FORMAT(1H+,83X,9HNEW AREA=,F8.3,5X,10HAREA LEFT=,F8.3)
      IF(DR(IA)) 36,37,39
   36 CALL EPS(10)
   37 IAB(I3,I4)=0
      GO TO 39
   38 CALL EPS(1)
   39 LN=3
      GO TO 1000
C                                     KO CARD
   40 WRITE(6,600) N2,CT(LR),KS,KT,N
      WRITE(6,604) (IX(J),J=1,2),(XX(J),J=1,2)
  604 FORMAT(1H+,23X,2I4,16X,2F10.3)
      IO=IX(1)
      IF(IO) 49,41,42
   49 LN=20
      GO TO 1000
   41 LN=2
      GO TO 1000
   42 IF(IO.GT.20) CALL EPS(1)
      IF((IX(2).LE.-1).OR.(IX(2).GT.10)) CALL EPS(1)
```

```
      LN=5
      GO TO 1000
C                                  PE CARD
   50 WRITE(6,600) N2,CT(LR),KS,KT,N
      WRITE(6,605) (IX(J),J=1,2),(XX(J),J=1,5)
  605 FORMAT(1H+,23X,2I4,16X,5F10.3)
      IP=IX(1)
      IF(IP.GT.0) GO TO 51
      LN=4
      GO TO 1000
   51 IF(IP.GT.120) GO TO 58
      IF(IX(2).GT.500) CALL EPS(1)
      AA=0.
      DO 54 J=1,4
      AR(J)=0.
      IB=IAB(IP,J)
      IF(IB.LE.0) GO TO 54
      AR(J)=DR(IB)
      AA=AA+AR(J)
   54 CONTINUE
      NP=NP+1
      WRITE(6,635) IP,AR
  635 FORMAT(55X,4HPART,I4,15H SURFACE AREAS=,4F10.3)
      IF(AA.LE.0.) CALL EPS(8)
      DO 55 J=1,2
      IF(XX(J+3).GT.AR(J)) CALL EPS(9)
   55 CONTINUE
      GO TO 59
   58 CALL EPS(1)
   59 LN=5
      GO TO 1000
C                                  RD CARD
   60 WRITE(6,600) N2,CT(LR)
      WRITE(6,606) IX(1),(XX(J),J=1,7)
  606 FORMAT(1H+,23X,I4,25X,7A6)
      DO 61 J=1,120
      DO 61 L=1,6
   61 IAB(J,K)=0
      DO 62 J=1,500
   62 DR(J)=0.
      DO 63 J=1,20
   63 LH(J)=0
      LN=7
      GO TO 1000
C                                  SD CARD
   70 KS=IX(1)
      WRITE(6,600) N2,CT(LR),KS
      WRITE(6,607) KS,(XX(J),J=1,7)
  607 FORMAT(1H+,23X,I4,25X,7A6)
      LN=8
      IF((KS.GT.0).AND.(KS.LE.20)) GO TO 1000
      CALL EPS(1)
      KS=-1
      GO TO 1000
C                                  TD CARD
   80 KT=IX(1)
      WRITE(6,600) N2,CT(LR),KS,KT
      WRITE(6,608) (IX(J),J=1,5),(XX(J),J=1,7)
  608 FORMAT(1H+,23X,5I4,9X,7A6)
      DO 81 J=1,20
   81 KKT(J)=0
```

```
      IF((KT.GT.0).AND.(KT.LE.100)) GO TO 82
      CALL EPS(1)
      KT=-1
   82 L1=IX(2)
      IF(L1.LE.0) GO TO 84
      IF(L1.GT.20) GO TO 83
      IF(LH(L1).LE.0) CALL EPS(2)
      L2=IX(3)
      IF(L2.LE.0) GO TO 84
      IF(L2.GT.20) GO TO 83
      IF(LH(L2).LE.0) CALL EPS(2)
      GO TO 84
   83 CALL EPS(1)
   84 L3=IX(4)
      IF(L3.LE.0) GO TO 86
      IF(L3.GT.20) GO TO 85
      LH(L3)=1
      GO TO 86
   85 CALL EPS(1)
   86 IF(IX(5).GT.20) CALL EPS(1)
      GO TO(801,802,803,804,805),IK
  801 LN=9
      GO TO 1000
  802 LN=12
      GO TO 1000
  803 LN=14
      GO TO 1000
  804 LN=16
      GO TO 1000
  805 LN=2
      GO TO 1000
C                                         EM CARD
   90 WRITE(6,600) N2,CT(LR),KS,KT
      WRITE(6,609) (XX(J),J=1,4)
  609 FORMAT(1H+,47X,4F10.3)
      LN=10
      GO TO 1000
C                                         ED CARD
  100 WRITE(6,600) N2,CT(LR),KS,KT
      WRITE(6,610) IX(1),(XX(J),J=1,4)
  610 FORMAT(1H+,23X,I4,25X,4A6)
      IF(IX(1).GT.0) GO TO 101
      LN=12
      GO TO 1000
  101 IF(IX(1).GT.10) CALL EPS(1)
      LN=11
      GO TO 1000
C                                         EQ CARD
  110 WRITE(6,600) N2,CT(LR),KS,KT
      WRITE(6,611) (IX(J),J=1,2),(XX(J),J=1,7)
  611 FORMAT(1H+,23X,2I4,16X,7F10.3)
      LN=10
      GO TO 1000
C                                         OD CARD
  120 WRITE(6,600) N2,CT(LR),KS,KT
      WRITE(6,612) IX(1),(XX(J),J=1,4)
  612 FORMAT(1H+,23X,I4,25X,4A6)
      IF(IX(1).GT.0) GO TO 121
      LN=14
      GO TO 1000
  121 IF(IX(1).GT.20) CALL EPS(1)
```

```
      LN=13
      GO TO 1000
C                                OQ CARD
  130 WRITE(6,600) N2,CT(LR),KS,KT
      WRITE(6,613) (IX(J),J=1,3)
  613 FORMAT(1H+,23X,3I4)
      LN=12
      GO TO 1000
C                                PD CARD
  140 WRITE(6,600) N2,CT(LR),KS,KT
      WRITE(6,614) IX(1),(XX(J),J=1,6)
  614 FORMAT(1H+,23X,I4,25X,6A6)
      IF(IX(1).GT.0) GO TO 141
      LN=16
      GO TO 1000
  141 IF(IX(1).GT.120) CALL EPS(1)
      IP=IX(1)
      LN=15
      GO TO 1000
C                                PQ CARD
  150 WRITE(6,600) N2,CT(LR),KS,KT
      WRITE(6,615) IX,(XX(J),J=1,2)
  615 FORMAT(1H+,23X,6I4,2F10.3)
      DO 152 J=1,4
      IF(IX(J).GT.500) GO TO 151
      IF((IX(J).GT.0).AND.(IX(J).LE.20)) GO TO 151
      IAB(IP,J)=IX(J)
      GO TO 152
  151 CALL EPS(1)
  152 CONTINUE
      IF(IX(5).GT.10) CALL EPS(1)
      IF(IX(6).GT.500) CALL EPS(1)
      IF(IX(6).LE.0) CALL EPS(3)
      LN=14
      GO TO 1000
C                                DD CARD
  160 WRITE(6,600) N2,CT(LR),KS,KT
      WRITE(6,616) (IX(J),J=1,2),(XX(J),J=1,4)
  616 FORMAT(1H+,23X,2I4,21X,4A6)
      JD=IX(1)
      M=IX(2)
      IF(IX(1).GT.0) GO TO 161
      LN=2
      GO TO 1000
  161 IF(IX(1).GT.500) CALL EPS(1)
      IF((M.GT.0).AND.(M.LE.11)) GO TO 162
      CALL EPS(1)
      M=11
  162 IF(ND.LT.IX(1)) ND=IX(1)
      LN=17
      GO TO 1000
C                                DQ CARD
  170 WRITE(6,600) N2,CT(LR),KS,KT
      WRITE(6,617) (XX(J),J=1,M)
  617 FORMAT(8X,F10.2,10F10.6)
      DR(JD)=XX(1)
      XR(JD)=XX(M+11)
      IF(M.EQ.1) GO TO 172
      SUM=0.
      DO 171 J=2,M
  171 SUM=SUM+XX(J)
```

```
      IF(ABS(SUM-1.).GT..0000001) CALL EPS(11)
  172 MP=M+11
      WRITE(6,637) (XX(J),J=12,MP)
  637 FORMAT(8X,11E10.2)
      LN=16
      IF(M.EQ.1) GO TO 1001
      DO 173 J=2,M
      IF(XX(J+10).LE.XX(J+11)) GO TO 173
      CALL EPS(6)
      GO TO 1001
  173 CONTINUE
      GO TO 1001
C                             ZD CARD
  180 WRITE(6,600) N2,CT(LR),KS,KT
      WRITE(6,618) IX(1),(XX(J),J=1,4)
  618 FORMAT(1H+,23X,I4,25X,4A6)
      IF(IX(1).GT.0) GO TO 181
      LN=1
      GO TO 1000
  181 IF(IX(1).GT.120) CALL EPS(1)
      LN=19
      GO TO 1000
C                             ZC CARD
  190 WRITE(6,600) N2,CT(LR),KS,KT
      WRITE(6,619) IX(1),(XX(J),J=1,4)
  619 FORMAT(1H+,23X,I4,20X,4F10.3)
      IF(IX(1).GT.0) GO TO 191
      LN=18
      GO TO 1000
  191 IF(IX(1).GT.120) CALL EPS(1)
      LN=19
      GO TO 1000
  200 WRITE(6,600) N2,CT(LR),KS,KT,N
      WRITE(6,605) (IX(J),J=1,2),(XX(J),J=1,4)
      IP=IX(1)
      IF(IP.GT.0) GO TO 201
      LN=4
      GO TO 1000
  201 IF(IP.GT.120) GO TO 208
      LK=IX(2)
      IF((LK.LE.0).OR.(LK.GT.500)) GO TO 58
      DO 202 J=1,4
      IF(XX(J)*XR(LK).GT.1.) CALL EPS(6)
  202 CONTINUE
      GO TO 209
  208 CALL EPS(1)
  209 LN=20
 1000 WRITE(9)  LR,IX,(XX(J),J=1,7)
      IF(LR.NE.LX) CALL EPS(7)
      RETURN
 1001 WRITE(9)  LR,XX
      IF(LR.NE.LX) CALL EPS(7)
      RETURN
      END
```

```
      SUBROUTINE EPS(LE)
C     ERROR PRINTING SUBROUTINE
      COMMON/T/LN,LR,IX(6),XX(22),M,N1,N2,I(6,20),X(22,20),K(22,20),
     .KIL,NE,KE(11),IAB(120,6),DR(500),XR(500),ND,IO,AR(4),KKT(20),NP,
     .LH(20),KXT                                                         A
      IF(LE.EQ.0) GO TO 100
      NE=NE+1
      KE(LE)=KE(LE)+1
      IF((KIL.EQ.1).AND.(LE.GT.7)) RETURN
      NP=NP+3
      IF(NP.GE.55) CALL PTS
      WRITE(6,600) NE
  600 FORMAT(/12H *****ERROR,I4)
      GO TO(1,2,3,4,5,6,7,8,9,10,11),LE
    1 WRITE(6,601)
  601 FORMAT(1H+,17X,30HINDEX OUT OF RANGE            /)
      RETURN
    2 WRITE(6,602)
  602 FORMAT(1H+,17X,30HPREREQUISITE MISSING          /)
      RETURN
    3 WRITE(6,603)
  603 FORMAT(1H+,17X,30HNECESSARY DATA MISSING        /)
      RETURN
    4 WRITE(6,604)
  604 FORMAT(1H+,17X,30HTOO MANY CARDS IN SET         /)
      RETURN
    5 WRITE(6,605)
  605 FORMAT(1H+,17X,30HINDEX USED PREVIOUSLY         /)
      RETURN
    6 WRITE(6,606)
  606 FORMAT(1H+,17X,30HDATA OUT OF RANGE             /)
      RETURN
    7 WRITE(6,607)
  607 FORMAT(1H+,17X,30HCARD OUT OF SEQUENCE          /)
      RETURN
    8 WRITE(6,608)
  608 FORMAT(1H+,17X,30HAFFECTED PART AREA ZERO       /)
      RETURN
    9 WRITE(6,609)
  609 FORMAT(1H+,17X,30HCONTACT AREA GT SURFACE AREA  /)
      RETURN
   10 WRITE(6,610)
  610 FORMAT(1H+,17X,30HREMAINING AREA NEGATIVE       /)
      RETURN
   11 WRITE(6,611)
  611 FORMAT(1H+,17X,30HTOTAL PROB. NOT EQUAL TO 1    /)
      RETURN
  100 WRITE(6,620)
  620 FORMAT(1H1//////20X,27HMICROBIAL BURDEN DATA CHECK/)
      IF(NE.GT.0) GO TO 150
      WRITE(6,621)
  621 FORMAT(24X,18HNO ERRORS THIS RUN)
      RETURN
  150 WRITE(6,622)
  622 FORMAT(20X,27H*****  ERROR SUMMARY  *****/)
      IOUT=0
      DO 152 J=1,7
      IF(KE(J)-1) 152,156,155
  155 WRITE(6,623) KE(J)
  623 FORMAT(I6,9H ERRORS -)
      GO TO 159
```

```
  156 WRITE(6,624) KE(J)
  624 FORMAT(I6,9H ERROR  -)
  159 IOUT=1
      GO TO(101,102,103,104,105,106,107),J
  101 WRITE(6,601)
      GO TO 152
  102 WRITE(6,602)
      GO TO 152
  103 WRITE(6,603)
      GO TO 152
  104 WRITE(6,604)
      GO TO 152
  105 WRITE(6,605)
      GO TO 152
  106 WRITE(6,606)
      GO TO 152
  107 WRITE(6,607)
  152 CONTINUE
      IF(KIL.NE.1) GO TO 154                                              A
      IF(KXT.EQ.1) RETURN                                                 A
      IF(IOUT.EQ.1) CALL EXIT                                             A
      RETURN                                                              A
  154 DO 153 J=8,11                                                       A
      IF(KE(J)-1) 153,158,157
  157 WRITE(6,623) KE(J)
      GO TO 160
  158 WRITE(6,624) KE(J)
  160 JM=J-7
      GO TO(108,109,110,111),JM
  108 WRITE(6,608)
      GO TO 153
  109 WRITE(6,609)
      GO TO 153
  110 WRITE(6,610)
      GO TO 153
  111 WRITE(6,611)
  153 CONTINUE
      IF(KXT.EQ.1) RETURN                                                 A
      IF(IOUT.EQ.1) CALL EXIT
      RETURN
      END
```

```
      SUBROUTINE PTS
C     PAGE TITLING SUBROUTINE
      COMMON/T/LN,LR,IX(6),XX(22),M,N1,N2,I(6,20),X(22,20),K(22,20),
     .KIL,NE,KE(11),IAB(120,6),DR(500),XR(500),ND,IO,AR(4),KKT(20),NP,
     .LH(20),KXT                                                        A
      WRITE(6,600)
  600 FORMAT( 115H1RECORD, TYPE, S/TSK/ST     1    2    3    4    5    6
     .7          8          9          10         11         12         13/)
      NP=0
      RETURN
      END
```

```
      SUBROUTINE BPS
C     BURDEN PREDICTION SUBROUTINE
C     THIS PROGRAM READS FROM TAPE 9
      COMMON KK,IK,KR,RUN(7),KS,STG(7),KT,TSK(7),DSC(6),ID,
     .NE,AES(4),IEC(10),IET(10),AEC(10),AET(10),AED(10),AEF(10,4),
     .NO,IOT(20),IOQ(20),IOC(20),
     .NP,DAC(120,2),IU(120),IAB(120,6),AAG(120),AAS(120),AAT(120),
     .ND,NX(572),DR(572,11),XR(572,11),KO,
     .NS,KKT(20),IR(5),AR(4),APA(2),
     .NT,JT(100),XMT(100),XVT(100),L1,L2,L3,L4,KTS,ITE
      DIMENSION XX(7),IX(6)
    3 FORMAT(34H1MICROBIAL BURDEN PREDICTION MODEL/4H RUN,I5,2H, ,7A6)
      REWIND 9
      REWIND 11
      NT=0
      DO 15 N=1,572
   15 NX(N)=-1
      DO 16 N=1,120                                                     A
      DO 16 J=1,6                                                       A
   16 IAB(N,J)=0                                                        A
  100 READ (9) L,KK,IK,I1,I2,I5,I6,XX
C     KK=1 INDICATES A NEW RUN, STAGE, AND TASK
C     KK=2 INDICATES A NEW STAGE AND TASK
C     KK=3 INDICATES A NEW TASK
C     KK=4 CALLS FOR A BURDEN DIFFERENCE DETERMINATION
C     KK=5 CALLS FOR ZONE BURDEN WRITEOUT
C        IK=0 CALLS FOR ZONE DEFINITION DATA INPUTS
C        IK=1 INDICATES ZONES CORRESPOND TO PARTS
C     KK=6 CALLS FOR A RESTART AND REQUIRES A BINARY DECK INPUT
      IF(KK.NE.6)GO TO 102
      REWIND 2                                                          A
      READ(2) KR,(RUN(J),J=1,7),KS,(STG(J),J=1,7),NE,(AES(J),J=1,4),    A
     . (IEC(J),IET(J),AEC(J),AET(J),AED(J),(AEF(J,K),K=1,4),J=1,NE),
     . NO,(IOT(J),IOQ(J),IOC(J),J=1,NO),NP,((DAC(J,K),K=1,2),IU(J),
     . (IAB(J,K),K=1,6),AAG(J),AAS(J),AAT(J),J=1,NP),ND,KO,NT,
     . (JT(J),XMT(J),XVT(J),J=1,NT),KTS,ID,ITE,K1,K2
  101 READ(2) N,NX(N),(DR(N,J),XR(N,J),J=1,11)                          A
      IF(N.LT.572) GO TO 101
      GO TO 100
  102 IF(KK-3) 103,130,109
  103 IF(NT.EQ.0)GO TO 106
C
C     WRITE STAGE SUMMARY FOR PRECEEDING STAGE
      WRITE(6,3)KR,(RUN(J),J=1,7)
      WRITE(6,4)KS,(STG(J),J=1,7)
    4 FORMAT(6H STAGE,I3,2H, ,7A6/15H STAGE SUMMARY-//
     .3X,4HTASK,7X,4HMEAN,6X,6HFINISH/13X,6HBURDEN,6X,4HTIME/)
      DO 104 I=1,NT
  104 WRITE(6,5)JT(I),XMT(I),XVT(I)
    5 FORMAT(I7,E13.3,F10.3)
  106 IF(KK.GT.0) GO TO 109
      IF(K2.EQ.3) GO TO 108
      WRITE(2) KR,(RUN(J),J=1,7),KS,(STG(J),J=1,7),NE,(AES(J),J=1,4),
     . (IEC(J),IET(J),AEC(J),AET(J),AED(J),(AEF(J,K),K=1,4),J=1,NE),
     . NO,(IOT(J),IOQ(J),IOC(J),J=1,NO),NP,((DAC(J,K),K=1,2),IU(J),
     . (IAB(J,K),K=1,6),AAG(J),AAS(J),AAT(J),J=1,NP),ND,KC,NT,
     . (JT(J),XMT(J),XVT(J),J=1,NT),KTS,ID,ITE,K1,K2
      N=1
  107 IF(NX(N).GE.0) WRITE(2) N,NX(N),(DR(N,J),XR(N,J),J=1,11)
      N=N+1
      IF(N.NE.572) GO TO 107
```

```
      WRITE(2) N,NX(1),(DR(1,J),XR(1,J),J=1,11)
  108 M=0
      WRITE(11) M,KR,KS,M,M,(DSC(J),J=1,4),M,M,(RUN(J),J=1,7),(STG(J),
     ,J=1,7),(JT(J),J=1,18)
      RETURN
  109 GO TO (110,120,130,140,150),KK
C
C     READ RUN NUMBER AND DESCRIPTION
  110 READ (9) L,KR,(IX(J),J=1,5),RUN
      WRITE(6,600) KR,(RUN(J),J=1,7)
  600 FORMAT(34H1MICROBIAL BURDEN PREDICTION MODEL//4H RUN,I2,2X,7A6//)
      ID=0
      MD=0
      K1=1
      K2=1
      NE=0
      NO=0
      NP=0
      ND=0
      NS=0
      NT=0
      DO 111 I=1,120
      IU(I)=0
      AAG(I)=0.
      AAS(I)=0.
      AAT(I)=0.
      DO 111 J=1,5
  111 IAB(I,J)=0
      DO 112 N=1,572
  112 NX(N)=-1
C
C     READ STAGE NUMBER AND DESCRIPTION
  120 READ (9) L,KS,(IX(J),J=1,5),STG
      NT=0
      M=1
      WRITE(11) M,KR,KS,M,M,(DSC(J),J=1,4),M,M,(RUN(J),J=1,7),(STG(J),
     ,J=1,7),(JT(J),J=1,18)
C
C     READ TASK NUMBER AND DESCRIPTION
  130 READ (9) L,KT,L1,L2,L3,L4,IX(1),TSK
      WRITE(6,612) KS,STG,KT,TSK
  612 FORMAT(6H1STAGE,I3,2X,7A6/5H TASK,I4,2X,7A6/)
C     L1 AND L2 ARE PREREQUISITE TIME DISTRIBUTIONS
C     L3 IS THE FINISH TIME DISTRIBUTION IF NEEDED
C        AS PREREQUISITE FOR ANOTHER TASK
C     L4 IS THE INDEX FOR SAVING THE TOTAL MICROBIAL BURDEN
C        IF NEEDED FOR USE IN DETERMINING A BURDEN DIFFERENCE
C
C     DETERMINE TASK START TIME
      KTS=0
      IF(L1.LE.0)GO TO 134
      L1=L1+530
      IF(L2.GT.0)GO TO 132
      KTS=L1
      GO TO 134
  132 L2=L2+530
      KTS=551
      CALL HCS(L1,L2,KTS,5)
      IF(ID.LE.0) GO TO 134
      IF(NX(KTS).GT.NX(ID)) CALL HAS(KTS)
  134 IF(I1.GT.0)K1=I1
```

```
      IF(I2.GT.0)K2=I2
      IF(I5.LE.0) GO TO 138
      IF(ID.GT.0) MD=NX(ID)
      ID=I5
C  RESTORE THE FIRST 20 DISTRIBUTIONS
      DO 137 I=1,20
      J=I+552
      IF(NX(J).LT.0) GO TO 137
      CALL HES(I,J)
  137 CONTINUE
  138 IF(I6.GT.0)ITE=I6
      NT=NT+1
      CALL MBS(MD,K1,K2)
      IF(K2.NE.2) GO TO 100
      REWIND 2
      WRITE(2) KR,(RUN(J),J=1,7),KS,(STG(J),J=1,7),NE,(AES(J),J=1,4),
     . (IEC(J),IET(J),AEC(J),AET(J),AED(J),(AEF(J,K),K=1,4),J=1,NE),
     . NO,(IOT(J),IOQ(J),IOC(J),J=1,NO),NP,((DAC(J,K),K=1,2),IU(J),
     . (IAB(J,K),K=1,6),AAG(J),AAS(J),AAT(J),J=1,NP),ND,KO,NT,
     . (JT(J),XMT(J),XVT(J),J=1,NT),KTS,ID,ITE,K1,K2
      N=1
  139 IF(NX(N).GE.0) WRITE(2) N,NX(N),(DR(N,J),XR(N,J),J=1,11)
      N=N+1
      IF(N.NE.572) GO TO 139
      WRITE(2) N,NX(1),(DR(1,J),XR(1,J),J=1,11)
      GO TO 100
C
C     DETERMINE AND WRITE BURDEN DIFFERENCE
  140 WRITE(6,3)KR,(RUN(J),J=1,7)
      WRITE(6,601)
  601 FORMAT(//18H BURDEN DIFFERENCE/)
      L1=I1+530
      L2=I2+530
      WRITE(6,602) I1,DR(L1,1)
  602 FORMAT(/17H BURDEN STORED IN,I3,7H, AREA=,F9.2,4X,6HBURDEN)
      CALL HWS(L1)
      WRITE(6,602) I2,DR(L2,1)
      CALL HWS(L2)
      A1=DR(L1,1)
      A2=DR(L2,1)
      DR(L1,1)=AVF(L1)
      DR(L2,1)=AVF(L2)
      CALL HCS(L2,L1,551,2)
      CALL HAS(551)
      WRITE(6,601)
      CALL HWS(551)
      DR(L1,1)=A1
      DR(L2,1)=A2
      GO TO 100
C
C     DETERMINE AND WRITE ZONE BURDEN DISTRIBUTIONS
 150  WRITE(6,3)KR,(RUN(J),J=1,7)
      WRITE(6,7)KS,KT
    7 FORMAT(43H MICROBIAL BURDEN BY ZONES (FOLLOWING STAGE,I3,6H, TASK,
     .I3,2H)-/)
      IF(IK.EQ.1)GO TO 180
C
C     USE ZONE DEFINITION INPUTS
  155 READ (9) L,IZ,(IX(J),J=1,5),DSC,XX(1)
      IF(IZ.LE.0)GO TO 100
      A=0.
```

```
      CALL HES(510,KO)
  160 READ (9) L,IP,(IX(J),J=1,5),AR,(XX(J),J=1,3)
C     IP IS THE PART
C     AR(J) THE FRACTION OF SURFACE J OF PART IP BELONGING TO ZONE IZ
      IF(IP.LE.0)GO TO 170
      DO 165 J=1,4
      IF(AR(J).EQ.0.)GO TO 165
      IF(IAB(IP,J).EQ.0)GO TO 165
      F=AR(J)
      IB=IAB(IP,J)
      CALL HMS(IB,F,509)
      A=A+DR(509,1)
      DR(509,1)=AVF(509)
      CALL HCS(510,509,510,1)
  165 CONTINUE
      GO TO 160
  170 WRITE(6,9)IZ,(DSC(J),J=1,4),A
    9 FORMAT(5H ZONE,I4,2X,4A6,7H, AREA=,F8.3/33X,11HZONE BURDEN)
      CALL HWS(510)
      GO TO 155
C
C     USE PARTS AS ZONES
  180 DO 190 I=1,NP
      IF(IU(I).EQ.0)GO TO 190
      A=0.
      CALL HES(510,KO)
      DO 185 J=1,4
      IB=IAB(I,J)
      IF(IB.LE.0) GO TO 185
      A=A+DR(IB,1)
      CALL HES(509,IB)
      DR(509,1)=AVF(IB)
      CALL HCS(510,509,510,1)
  185 CONTINUE
      WRITE(6,10)I,DAC(I,1),DAC(I,2),DR(510,1)
   10 FORMAT(5H ZONE,I4,2X,2A6,7H, AREA=,F7.2,8H, BURDEN)                  A
      CALL HWS(510)
  190 CONTINUE
      GO TO 100
      END
```

```
      SUBROUTINE MBS(MD,K1,K2)
C     MICROBIAL BUILDUP SUBROUTINE
      COMMON KK,IK,KR,RUN(7),KS,STG(7),KT,TSK(7),DSC(6),ID,
     .NE,AES(4),IEC(10),IET(10),AEC(10),AET(10),AED(10),AEF(10,4),
     .NO,IOT(20),IOQ(20),IOC(20),
     .NP,DAC(120,2),IU(120),IAB(120,6),AAG(120),AAS(120),AAT(120),
     .ND,NX(572),DR(572,11),XR(572,11),KO,
     .NS,KKT(20),IR(5),AR(4),APA(2),
     .NT,JT(100),XMT(100),XVT(100),L1,L2,L3,L4,KTS,ITE
      DIMENSION IX(6),XX(7)
      GO TO (100,120,140,160,200),IK
C
C     ENVIRONMENTS INPUTS-
  100 READ (9) L,IX,AES,(XX(J),J=1,3)
      WRITE(6,8)(AES(J),J=1,4)
    8 FORMAT(//30H ENVIRONMENTS INPUTS - - - - -/5H AES=,4E12.4)
C     AES(J) IS THE SURFACE LIFETIME MODIFIER FOR SURFACE J
      DO 110 I=1,10
      READ (9) L,N,(IX(J),J=1,5),DSC,XX(1)
      IF(N.LE.0)GO TO 120
      READ (9) L,IEC(N),IET(N),(IX(J),J=1,4),AEC(N),AET(N),AED(N),
     .(AEF(N,J),J=1,4)
C     IEC,AEC DESCRIBE THE ENVIRONMENTS BIOTA CONCENTRATION
C     IET,AET DESCRIBE THE REFERENCE ACCRETION TIME
C     AED IS THE AIRBOURNE CONCENTRATION PER MAN
C     AEF IS THE RATE AT WHICH, FOR ALL OTHER FACTORS STANDARD,
C        BIOTA REACH EACH SURFACE
      WRITE(6,20)N,(DSC(J),J=1,4),IEC(N),AEC(N),IET(N),AET(N),AED(N),
     .(AEF(N,J),J=1,4)
   20 FORMAT(/I4,1X,4A6,4X,4HIEC=,I3,6H, AEC=,F7.4,6H, IET=,I3,6H, AET=,
     .F7.2,6H, AED=,F7.4,6H, AEF=,4F6.2)
  110 IF(NE.LT.N)NE=N
      READ (9) L,N,(IX(J),J=1,5),DSC,XX(1)
C
C     OPERATIONS INPUTS-
  120 WRITE(6,9)
    9 FORMAT(//30H OPERATIONS INPUTS - - - - - -/)
      DO 130 I=1,20
      READ (9) L,N,(IX(J),J=1,5),DSC,XX(1)
      IF(N.LE.0)GO TO 140
      READ (9) L,IOT(N),IOQ(N),IOC(N),(IX(J),J=1,3),XX
      WRITE(6,21)N,(DSC(J),J=1,4),IOT(N),IOQ(N),IOC(N)
   21 FORMAT(/I4,1X,4A6,4X,4HIOT=,I3,6H, IOQ=,I3,6H, IOC=,I3)
C     IOT IS THE OPERATION TIME INTERVAL
C     IOQ IS THE DIRTINESS FACTOR
C     IOC IS THE BIOTA CONCENTRATION FOR CONTACT CONTAMINATION
  130 IF(NO.LT.N)NO=N
      READ (9) L,N,(IX(J),J=1,5),DSC,XX(1)
C
C     PARTS INPUTS-
  140 WRITE(6,10)
   10 FORMAT(//30H PARTS INPUTS  - - - - - - - -/)
      DO 150 I=1,120
      READ (9) L,N,(IX(J),J=1,5),DSC,XX(1)
      IF(N.LE.0)GO TO 160
      DAC(N,1)=DSC(5)
      DAC(N,2)=DSC(6)
      IU(N)=0
      AAT(N)=0.
      IF(KTS.GT.0)AAT(N)=DR(KTS,1)
      READ (9) L,(IAB(N,J),J=1,6),AAG(N),AAS(N),(XX(J),J=1,5)
```

```
      IF(IAB(N,6).GT.0)GO TO 142
      AAG(N)=0.
      AAS(N)=0.
  142 CONTINUE
C     DAC IS THE PERMANENT ALPHAMERIC DESCRIPTION OF PART N
C     IAB(J) INDICATES THE DISTRIBUTION FOR AREA/BURDEN FOR
C        J=1, TOP SURFACE
C        J=2, OTHER EXTERIOR SURFACE
C        J=3, MATED SURFACE
C        J=4, OCCLUDED SURFACE
C     IAB(5) IS THE ENVIRONMENT INDEX (MAY BE LEFT BLANK)
C     IAB(6) IS THE RETENTION DISTRIBUTION FOR CONTACT
C     AAG IS THE RETENTION DISTRIBUTION MEAN FOR FALLOUT
C     AAS IS THE RETENTION DISTRIBUTION MEAN FOR CONTACT
C     AAT IS THE LAST TIME OF ACCRETION UPDATE
      WRITE(6,22)N,(DSC(J),J=1,6),(IAB(N,J),J=1,6),AAG(N),AAS(N)
   22 FORMAT(/I4,1X,6A6,4X,4HIAB=,6I4,6H, AAG=,F7.3,6H, AAS=,F7.3)
  150 IF(NP.LT.N)NP=N
      READ (9) L,N,(IX(J),J=1,5),DSC,XX(1)
C
C     DISTRIBUTIONS INPUTS -
  160 WRITE(6,11)
   11 FORMAT(//30H DISTRIBUTIONS INPUTS  - - - -//)
      DO 175 I=1,500
      READ (9) L,N,M,(IX(J),J=1,4),DSC,XX(1)
C     M=1 INDICATES THE DISTRIBUTION IS A CONSTANT
      IF(N.LE.0)GO TO 179
      READ (9) L,(DR(N,J),J=1,11),(XR(N,J),J=1,11)
      NX(N)=M
      IF(DR(N,1).LE.0.) DR(N,1)=AVF(N)
      IF(N.GT.20) GO TO 171
      J=N+552
      CALL HES(J,N)
  171 IF(ID.LE.0) GO TO 172
      IF(M.GT.NX(ID)) CALL HAS(N)
  172 WRITE(6,23)N,(DSC(J),J=1,4),DR(N,1)
   23 FORMAT(I4,1X,4A6/5X,12HCOEFFICIENT=,E11.4,6X,12HDISTRIBUTION)
      CALL HWS(N)
      IF(ND.LT.N) ND=N
  175 CONTINUE
      READ (9) L,N,M,(IX(J),J=1,4),DSC,XX(1)
  179 IF(ID.LE.0) GO TO 190
      IF(MD.EQ.NX(ID)) GO TO 190
      DO 180 I=1,20
      IF(NX(I).GT.NX(ID)) CALL HAS(I)
  180 CONTINUE
      IF(MD.LE.NX(ID)) GO TO 189
      DO 185 I=21,552
      IF(NX(I).GT.NX(ID)) CALL HAS(I)
  185 CONTINUE
  189 MD=NX(ID)
C
C     THE ZERO DISTRIBUTION
  190 IF(ND.GE.500) WRITE(6,630)
  630 FORMAT(///33H **** TOO MANY DISTRIBUTIONS ****///)
      ND=ND+1
      KO=ND
      NX(KO)=1
      DR(KO,1)=0.
      XR(KO,1)=0.
C     SUBTASK LEVEL MICROBIAL BUILDUP DETERMINATION
```

```
 200   IF(KTS.EQ.0)KTS=KO
C      KTS IS THE TASK START TIME
       LL=KTS
       NS=0
       DO 295 I=1,20
       READ (9) L,N,N1,N2,(IX(J),J=1,3),DSC,XX(1)
C      N1 AND N2 ARE PREREQUISITE SUBTASKS
       IF(N.LE.0)GO TO 300
       WRITE(6,12)N,(DSC(J),J=1,4)
  12   FORMAT(/8H SUBTASK,I3,2H, ,4A6)
       IF(NS.LT.N)NS=N
       IF(N1.GT.0)GO TO 205
       ITT=KTS
       GO TO 215
 205   J1=KKT(N1)
       IF(N2.GT.0)GO TO 210
       ITT=J1
       GO TO 215
  210  J2=KKT(N2)
       ITT=510
       CALL HCS(J1,J2,ITT,5)
C      ITT IS THE SUBTASK START TIME INDEX
 215   T1=DR(ITT,1)
       M=2
       WRITE(11) M,KR,KS,KT,N,(DSC(J),J=1,4),ITT,NX(ITT),(DR(ITT,J),J=1,
      .11),(XR(ITT,J),J=1,11),(AEC(J),J=1,5),N1,N2,(JT(J),J=1,3)
       NC=0
C      ENVIRONMENT,AREA/BIOTA BURDEN CHANGES-
  220  READ (9) L,K,IR,AR,(XX(J),J=1,3)
C      K IS THE CHANGE IDENTIFIER
C      IR(1),IR(2) ARE THE NEW PART AND SURFACE INDICES
C      IR(3),IR(4) ARE THE CONTRIBUTING PART AND SURFACE
C      IR(5) IS THE NEW ENVIRONMENT (IF ANY) FOR PART IR(1)
C      AR(1) IS THE AREA CHANGED TO IR(1),IR(2) FROM IR(3),IR(4)
C      AR(2) IS THE NEW AAG FACTOR
C      AR(3) IS THE NEW AAS FACTOR
C      IF IR(5), AR(1), AR(2), AR(3) ZERO, NO CHANGE IS MADE
       IF(K.LE.0)GO TO 240
       NC=NC+1
       IF(NC.EQ.1)WRITE(6,16)
  16   FORMAT(/6X,42HENVIRONMENT/AREA/RETENTION FACTOR CHANGES-/)
       I1=IR(1)
       IF(NP.LT.I1)NP=I1
       IF(AR(1).LE.0.)GO TO 232
       CALL MAS(I1,T1)
       IU(I1)=IU(I1)+1
       I2=IR(2)
       I3=IR(3)
       CALL MAS(I3,T1)
       IU(I3)=IU(I3)+1
       I4=IR(4)
       IB=IAB(I1,I2)
       IA=IAB(I3,I4)
       IF(IB.GT.0)GO TO 222
       IF(ND.GE.500) WRITE(6,630)
       ND=ND+1
       IB=ND
       IAB(I1,I2)=IB
       CALL HES(IB,KO)
 222   IC=ND+1
       IF(IR(5).GT.0) IAB(I1,5)=IR(5)
```

```
      IF(AR(2).GT.0.) AAG(I1)=AR(2)
      IF(AR(3).GT.0.) AAS(I1)=AR(3)
      IF(DR(IA,1).LE.0.)GO TO 220
      F=AR(1)/DR(IA,1)
      IF(F.LT..995)GO TO 230
      F=1.
      IAB(I3,I4)=0
 230  CALL HMS(IA,F,IC)
      A=DR(IB,1)+DR(IC,1)
      DR(IB,1)=AVF(IB)
      DR(IC,1)=AVF(IC)
      CALL HCS(IB,IC,IB,1)
      DR(IB,1)=A
      F=1.-F
      CALL HMS(IA,F,IA)
      GO TO 220
  232 IF(IR(5).GT.0) IAB(I1,5)=IR(5)
      IF(AR(2).GT.0.) AAG(I1)=AR(2)
      IF(AR(3).GT.0.) AAS(I1)=AR(3)
      GO TO 220
C
C     OPERATIONAL LEVEL BUILDUP-
 240  IT=510+N
      CALL HES(IT,ITT)
C     ITT IS THE SUBTASK START TIME DISTRIBUTION INDEX
C     IT=510+N IS THE SUBTASK CURRENT TIME DISTRIBUTION INDEX
      KKT(N)=IT
      IE=ITE
  250 READ (9) L,IO,IKE,(IX(J),J=1,4),AKT,AKQ,(XX(J),J=1,5)
C     IO IS THE OPERATION
C     IKE IS THE OPERATION ENVIRONMENT WHEN DIFFERENT FROM
C        THE TASK ENVIRONMENT OR THE PREVIOUS OPERATION ENVIRONMENT
C     AKT IS THE OPERATION TIME MODIFIER
C     AKQ IS THE NUMBER OF MEN
C        IO LT 0 INDICATES A DECONTAMINATION OPERATION FOR WHICH
C        AKT IS THE FIXED OPERATION INTERVAL
      IF(IO)252,290,258
C
C     DECONTAMINATION
 252  IO=-IO
      T1=DR(IT,1)
      IF(AKT.LE.0.)GO TO 253
      I7=ND+7
      NX(I7)=1
      DR(I7,1)=AKT
      XR(I7,1)=AKT
      CALL HCS(IT,I7,IT,1)
  253 IF(K1.GE.2) WRITE(6,37)IO,T1,DR(IT,1)
   37 FORMAT(5X,9HOPERATION,I3,25H, (DECONTAMINATION), FROM,F7.2,3H TO,
     .F7.2,6H HOURS)
      T1=DR(IT,1)                                                      A
  254 READ (9) L,IP,LK,(IX(J),J=1,4),AR,(XX(J),J=1,3)
C     IP IS THE PART AFFECTED
C     LK IS THE CURVE DESCRIBING FRACTION OF BIOTA REMOVED
C     AR(J) IS THE MEAN FRACTION OF BIOTA REMOVED FROM SURFACE J
      IF(IP.LE.0)GO TO 250
      IU(IP)=IU(IP)+1
      DT=T1-AAT(IP)
      CALL MAS(IP,T1)
      DO 256 J=1,4
      IF(AR(J).LE.0.)GO TO 256
```

```
      IB=IAB(IP,J)
      IF(IB.LE.0)GO TO 256
      A=DR(IB,1)
      IF(A.LE.0.) GO TO 256
      IF(DT.LE.0.) GO TO 255
      M=8
      WRITE(11) M,KR,KS,KT,N,(DSC(L),L=1,4),IB,NX(IB),(DR(IB,L),L=1,11),
     .(XR(IB,L),L=1,11),A,T1,(DAC(IP,L),L=1,2),A,IP,J,IO,IAB(IP,5),IT
  255 F=(1.-AR(J))/DR(LK,1)                                            A
      IC=ND+1
      CALL HMS(LK,F,IC)
      DR(IB,1)=AVF(IB)
      DR(IC,1)=AVF(IC)
      CALL HCS(IB,IC,IB,3)
      BD=DR(IB,1)
      DR(IB,1)=A
      M=9
      WRITE(11) M,KR,KS,KT,N,(DSC(L),L=1,4),IB,NX(IB),(DR(IB,L),L=1,11),
     .(XR(IB,L),L=1,11),A,T1,AKT,AR(J),A,IP,J,IO,LK,IP
      IF(K1.EQ.3) WRITE(6,39)IP,DAC(IP,1),DAC(IP,2),J,A,AR(J),BD
   39 FORMAT(10X,4HPART,I4,2X,2A6,9H, SURFACE,I2,7H, AREA=,F8.3,
     .19H, FRACTION REMOVED=,F5.3,17X,9H, BURDEN=,E10.3)               A
  256 CONTINUE
      GO TO 254
C
C     FALLOUT CONTAMINATION-
  258 IF(IKE.GT.0)IE=IKE
      I2=ND+2
      I3=ND+3
      I4=ND+4
      I5=ND+5
      I6=ND+6
      I7=ND+7
      IC=IOC(IO)
      IQ=IOQ(IO)
      II=IET(IE)
      JC=IEC(IE)
      ITK=IOT(IO)
      T1=DR(IT,1)
      IF(ITK.EQ.0)GO TO 259
      F=AKT/DR(ITK,1)
      CALL HMS(ITK,F,I7)
      DR(I7,1)=AVF(I7)
      CALL HCS(IT,I7,IT,1)
C     I7 INDICATES THE OPERATION TIME DISTRIBUTION
  259 IF(K1.GE.2) WRITE(6,29) IO,IE,T1,DR(IT,1),AKQ
   29 FORMAT(5X,9HOPERATION,I3,13H, ENVIRONMENT,I3,6H, FROM,F7.2,3H TO,
     .F7.2,7H HOURS,,F3.0,4H MEN)
      AQ=AKQ/DR(IQ,1)
      Q = AQ*AED(IE)
      CALL HMS(IQ,Q,I5)
      DR(I5,1)=AVF(I5)
C     I5 IS THE CURVE Q*D
      AC=AEC(IE)/DR(JC,1)
      CALL HMS(JC,AC,I6)
      DR(I6,1)=AVF(I6)
C     I6 IS THE CURVE C
      CALL HCS(I5,I6,I6,1)
C     I6 INDICATES THE TOTAL FALLOUT SOURCE CONCENTRATION (C+Q*D)
  260 READ (9) L,IP,LS,(IX(J),J=1,4),APD,APC,APS,APA,(XX(J),J=1,2)
C     IP IS THE PART AFFECTED
```

```
C     LS IS THE TOOL STICKINESS DISTRIBUTION
C     APD IS NOT USED IN THIS VERSION
C     APC IS THE TOOL BIOTA CONCENTRATION MODIFIER
C     APS IS THE MEAN TOOL STICKINESS
C     APA IS THE CONTACT AREA FOR EACH TOUCHED SURFACE
      IF(IP.LE.0)GO TO 250
      IU(IP)=IU(IP)+1
      DT=T1-AAT(IP)
      CALL MAS(IP,T1)
      IF(AQ.LE.0.)GO TO 275
      IF(IE.LE.0)GO TO 275
      IF(ITK.LE.0)GO TO 275
      AAT(IP)=DR(IT,1)
      DO 270 J=1,4
      IB=IAB(IP,J)
      F=AEF(IE,J)*AAG(IP)
      IF(IB.LE.0)GO TO 270
      IF(F.LE.0.)GO TO 270
      A=DR(IB,1)
      IF(A.LE.0.)GO TO 270
      IF(DT.LE.0.) GO TO 263
      M=8
      WRITE(11) M,KR,KS,KT,N,(DSC(L),L=1,4),IB,NX(IB),(DR(IB,L),L=1,11),
     .(XR(IB,L),L=1,11),A,T1,(DAC(IP,L),L=1,2),A,IP,J,IO,IAB(IP,5),IT
  263 F=F*A
      CALL HMS(I6,F,I5)
C     I5 IS THE CURVE A*R = A*F*G*(C+Q*D)
      EE=AET(IE)*AES(J)/DR(II,1)
      CALL HMS(II,EE,I2)
      DR(I2,1)=AVF(I2)
C     I2 IS THE CURVE V
      CALL HCS(I5,I2,I4,3)
C     I4 IS THE CURVE A*V*R
      CALL HCS(I7,I2,I2,4)
C     I2 IS THE CURVE T/V
      M=NX(I2)
      NX(I3)=M
      DO 265 JJ=1,M
      DR(I3,JJ)=DR(I2,JJ)
      XR(I2,JJ)=EXP(-XR(I2,JJ))
  265 XR(I3,JJ)=1.-XR(I2,JJ)
      DR(I2,1)=AVF(I2)
      DR(I3,1)=AVF(I3)
C     I2 IS THE CURVE EXP(-T/V)
C     I3 IS THE CURVE 1-EXP(-T/V)
      CALL HCS(I4,I3,I4,3)
      DR(IB,1)=AVF(IB)
      CALL HCS(IB,I2,I2,3)
      CALL HCS(I2,I4,IB,1)
      CALL HAS(IB)
      BD=DR(IB,1)
      DR(IB,1)=A
      M=3
      WRITE(11) M,KR,KS,KT,N,(DSC(L),L=1,4),IB,NX(IB),(DR(IB,L),L=1,11),
     .(XR(IB,L),L=1,11),A,T1,AKT,AKQ,A,IP,J,IO,IE,IP
      IF(K1.EQ.3) WRITE(6,30) IP,DAC(IP,1),DAC(IP,2),J,A,APC,APA(J),BD
   30 FORMAT(10X,4HPART,I4,2X,2A6,9H, SURFACE,I2,7H, AREA=,F8.3,
     .14H, TOOL BURDEN=,F6.0,15H, AREA TOUCHED=,F6.3,9H, BURDEN=,E10.3)
  270 CONTINUE
C
C     CONTACT CONTAMINATION-
```

A

```
 275  IF(IC.LE.0)GO TO 260
      DR(IC,1)=AVF(IC)
      IG=IAB(IP,6)
      IF(IG.LE.0) GO TO 260
      DR(IG,1)=AVF(IG)
      S1=.5*AAS(IP)*APC
      IF(LS.LE.0)GO TO 260
      DR(LS,1)=AVF(LS)
      IF(DR(LS,1).LE.0.)GO TO 260
      S2=.5*APS/DR(LS,1)
      B=S1*S2
      IF(B.EQ.0.)GO TO 260
      S1=S1/(DR(IC,1)*DR(IG,1))
      DO 280 J=1,2
      IB=IAB(IP,J)
      IF(IB.LE.0)GO TO 280
      A=DR(IB,1)
      DR(IB,1)=AVF(IB)
      IF(A.LE.0.)GO TO 280
      IF(APA(J).LE.0.)GO TO 280
      F=S1*APA(J)
      CALL HMS(IC,F,I4)
      CALL HCS(I4,IG,I4,3)
      F=S2*APA(J)/A
      CALL HMS(LS,F,I5)
      CALL HCS(IB,I5,I5,3)
      CALL HCS(IB,I5,IB,2)
      CALL HCS(IB,I4,IB,1)
      CALL HAS(IB)
      DR(IB,1)=A
      M=4
      WRITE(11) M,KR,KS,KT,N,(DSC(L),L=1,4),IB,NX(IB),(DR(IB,L),L=1,11),
     .(XR(IB,L),L=1,11),A,APA(J),AAS(IP),APS,APC,IP,J,IAB(IP,6),IC,IP
 280  CONTINUE
      GO TO 260
 290  IF(DR(LL,1).LT.DR(IT,1))LL=IT
 295  CONTINUE
      READ (9) L,N,N1,N2,(IX(J),J=1,3),DSC,XX(1)
 300  IF(L3.LE.0)GO TO 302
      L3=L3+530
      CALL HES(L3,LL)
  302 CONTINUE
      T=DR(LL,1)
      WRITE(6,601) KR,KS,KT,(TSK(J),J=1,7)
  601 FORMAT(13H1TASK SUMMARY/4H RUN,I2,7H, STAGE,I3,6H, TASK,I4,2X,7A6,
     .//29X,15HTASK START TIME)
      CALL HWS(KTS)
      WRITE(6,604)
  604 FORMAT(29X,16HTASK FINISH TIME)
      CALL HWS(LL)
      WRITE(6,605)
  605 FORMAT(/36H BURDEN DISTRIBUTIONS AT END OF TASK//50H BURDENS BY ZO
     .NE AND SURFACE   - - - - - - - - - - -/)
      DO 303 J=1,4
      JJ=500+J
      CALL HES(JJ,KO)
  303 AR(J)=0.
      DO 310 I=1,NP
      IF(IU(I).EQ.0)GO TO 310
      CALL MAS(I,T)
      WRITE(6,602) I,(DAC(I,J),J=1,2)
```

```
 602 FORMAT(/5H ZONE,I4,2X,2A6)
     DO 305 J=1,4
     JJ=500+J
     IB=IAB(I,J)
     IF(IB.LE.0)GO TO 305
     IF(DR(IB,1).LE.0.)GO TO 305
     A=DR(IB,1)
     AR(J)=AR(J)+A
     DR(IB,1)=AVF(IB)
     CALL HCS(JJ,IB,JJ,1)
     DR(IB,1)=A
     WRITE(6,603) J,DR(IB,1)
 603 FORMAT(5X,14HSURFACE NUMBER,I2,7H, AREA=,F9.3,8H, BURDEN)
     CALL HWS(IB)
305  CONTINUE
310  CONTINUE
     WRITE(6,608)
 608 FORMAT(//50H BURDEN TOTALS BY SURFACE  - - - - - - - - - - - - - -//)
     A=0.
     CALL HES(505,K0)
     DO 320 J=1,4
     JJ=500+J
     CALL HAS(JJ)
     WRITE(6,606) J,AR(J)
 606 FORMAT(/8H SURFACE,I2,13H, TOTAL AREA=,F9.3,14H, TOTAL BURDEN)
     CALL HWS(JJ)
     A=A+AR(J)
320  CALL HCS(505,JJ,505,1)
     CALL HAS(505)
     WRITE(6,609)
 609 FORMAT(//50H - - - - - - - - - - - - - - - - - - - - - - - - -//)
     WRITE(6,607) A
 607 FORMAT(//28H TOTAL, ALL SURFACES.  AREA=,F9.3,9H,  BURDEN)
     CALL HWS(505)
     JT(NT)=KT
     XMT(NT)=AVF(505)
     XVT(NT)=AVF(LL)
     IF(L4.LE.0)RETURN
     L4=L4+530
     CALL HES(L4,505)
     DR(L4,1)=A
     RETURN
     END
```

```
      SUBROUTINE HCS(IA,IB,IC,K)
C
C     HISTOGRAM COMBINING SUBROUTINE
      DIMENSION CA(11),CB(11),CR(21),ZR(21),QR(11,11),NQ(11)
      COMMON KK,IK,KR,RUN(7),KS,STG(7),KT,TSK(7),DSC(6),ID,
     .NE,AES(4),IEC(10),IET(10),AEC(10),AET(10),AED(10),AEF(10,4),
     .NO,IOT(20),IOQ(20),IOC(20),
     .NP,DAC(120,2),IU(120),IAB(120,6),AAG(120),AAS(120),AAT(120),
     .ND,NX(572),DR(572,11),XR(572,11),KO,
     .NS,KKT(20),IR(5),AR(4),APA(2),
     .NT,JT(100),XMT(100),XVT(100),L1,L2,L3,L4,KTS,ITE
      NA=NX(IA)
      NB=NX(IB)
      A=DR(IA,1)
      B=DR(IB,1)
      X=XR(IA,1)
      Y=XR(IB,1)
      CR(1)=ZF(A,B,K)
      CM=CR(1)
      DR(IC,1)=CR(1)
      ZR(1)=ZF(X,Y,K)
      IF(NA.GT.1)GO TO 50
      IF(X.NE.0.)GO TO 30
      IF((K-3)*(K-4).EQ.0)GO TO 35
   30 IF(NB.GT.1)GO TO 40
   35 XR(IC,1)=ZR(1)
      NX(IC)=1
      GO TO 1000
   40 NC=NB
      DO 45 J=2,NC
      Y=XR(IB,J)
      CR(J)=DR(IB,J)
   45 ZR(J)=ZF(X,Y,K)
      GO TO 60
   50 IF(NB.GT.1)GO TO 90
      NC=NA
      DO 55 J=2,NC
      X=XR(IA,J)
      CR(J)=DR(IA,J)
   55 ZR(J)=ZF(X,Y,K)
   60 NX(IC)=NC
      IF(ZR(NC).GT.ZR(1))GO TO 70
      XR(IC,1)=ZR(NC)
      JC=NC
      DO 65 J=2,NC
      DR(IC,J)=CR(JC)
      JC=JC-1
   65 XR(IC,J)=ZR(JC)
      GO TO 80
   70 XR(IC,1)=ZR(1)
      DO 75 J=2,NC
      DR(IC,J)=CR(J)
   75 XR(IC,J)=ZR(J)
   80 IF(K.LE.4)GO TO 1000
C     ELIMINATION OF DUPLICATE X VALUES FOR K=5
      CR(1)=0.
      CR(2)=0.
      JC=1
      ZR(1)=XR(IC,1)
      DO 85 J=2,NC
      CR(J)=CR(J)+DR(IC,J)
```

```
      IF(XR(IC,J).LE.XR(IC,J-1))GO TO 85
      JC=JC+1
      CR(JC)=CR(J)
      ZR(JC)=XR(IC,J)
   85 CR(J+1)=CR(J)
      NC=JC
      GO TO 300
   90 IF(K.EQ.5)GO TO 200
      JA=0
      JB=0
      JC=0
      KA=NA+1
      KB=NB+1
      IF(NA.GE.NB)GO TO 94
      NC=NB
      KC=KB
      GO TO 100
   94 NC=NA
      KC=KA
  100 JC=JC+1
      KC=KC-1
      IF(JC.GT.KC)GO TO 140
      IF(JA+1.GE.KA)GO TO 124
      JA=JA+1
      KA=KA-1
      X1=XR(IA,JA)
      X2=XR(IA,KA)
      IF(JB+1.GE.KB)GO TO 120
      JB=JB+1
      KB=KB-1
      Y1=XR(IB,JB)
      Y2=XR(IB,KB)
      IF(K.EQ.2)GO TO 110
      Z1=ZF(X1,Y1,K)
      Z2=ZF(X1,Y2,K)
      IF(Z2.GT.Z1)GO TO 102
      ZR(JC)=Z2
      GO TO 104
  102 ZR(JC)=Z1
  104 Z1=ZF(X2,Y1,K)
      Z2=ZF(X2,Y2,K)
      IF(Z2.GT.Z1)GO TO 106
      ZR(KC)=Z1
      GO TO 100
  106 ZR(KC)=Z2
      GO TO 100
  110 ZR(JC)=ZF(X1,Y1,K)
      ZR(KC)=ZF(X2,Y2,K)
      IF(ZR(JC)-ZR(KC))100,112,112
  112 ZR(JC)=.5*(ZR(JC)+ZR(KC))
  114 KC=KC+1
      IF(KC.GT.NC)GO TO 116
      JC=JC+1
      ZR(JC)=ZR(KC)
      GO TO 114
  116 NC=JC
      GO TO 140
  120 Y1=.5*(XR(IB,JB)+XR(IB,KB))
      ZR(JC)=ZF(X1,Y1,K)
      ZR(KC)=ZF(X2,Y1,K)
      GO TO 100
```

```
124   X1=.5*(XR(IA,JA)+XR(IA,KA))
      IF(JB+1.GE.KB)GO TO 128
      JB=JB+1
      KB=KB-1
      Y1=XR(IB,JB)
      Y2=XR(IB,KB)
      IF(K.EQ.2)GO TO 130
      Z1=ZF(X1,Y1,K)
      Z2=ZF(X1,Y2,K)
      IF(Z2.GT.Z1)GO TO 126
      ZR(JC)=Z2
      ZR(KC)=Z1
      GO TO 100
126   ZR(JC)=Z1
      ZR(KC)=Z2
      GO TO 100
128   Y1=.5*(XR(IB,JB)+XR(IB,KB))
      ZR(JC)=ZF(X1,Y1,K)
      ZR(KC)=ZR(JC)
      GO TO 100
130   ZR(JC)=ZF(X1,Y2,K)
      ZR(KC)=ZF(X1,Y1,K)
      GO TO 100
140   NX(IC)=NC
      IF(NC.EQ.1)GO TO 35
      IF(K.EQ.2)GO TO 171
      DO 150 J=1,NC
150   CR(J)=0.
      DO 170 JA=2,NA
      DO 170 JB=2,NB
      P=DR(IA,JA)*DR(IB,JB)
      X1=XR(IA,JA-1)
      X2=XR(IA,JA)
      Y1=XR(IB,JB-1)
      Y2=XR(IB,JB)
      Z1=ZF(X1,Y1,K)
      Z2=ZF(X1,Y2,K)
      IF(Z2.GT.Z1)GO TO 154
      ZA=Z2
      GO TO 155
154   ZA=Z1
155   Z1=ZF(X2,Y1,K)
      Z2=ZF(X2,Y2,K)
      IF(Z2.GT.Z1)GO TO 158
      ZB=Z1
      GO TO 160
158   ZB=Z2
160   DO 170 JC=2,NC
      IF(ZR(JC).LE.ZA)GO TO 170
      IF(ZR(JC).GE.ZB)GO TO 166
      CR(JC)=CR(JC)+P*(ZR(JC)-ZA)/(ZB-ZA)
      GO TO 170
166   CR(JC)=CR(JC)+P
 170  CONTINUE
      GO TO 300
171   KC=1
      DO 172 JC=2,NC
      IF(ZR(JC).GT.ZR(NC))ZR(JC)=ZR(NC)
      IF(ZR(KC).GE.ZR(JC))GO TO 172
      KC=KC+1
      ZR(KC)=ZR(JC)
```

```
172   CONTINUE
      NC=KC
      IF(NC.GT.1)GO TO 173
      NX(IC)=1
      XR(IC,1)=ZR(1)
      GO TO 1000
173   DO 174 JA=2,NA
174   CA(JA)=DR(IA,JA)
      DO 180 JC=2,NC
      Z1=ZR(JC-1)
      Z2=ZR(JC)
      DO 175 JA=1,NA
175   QR(JA,JC)=0.
      DO 178 JB=2,NB
      Y1=XR(IB,JB-1)
      Y2=XR(IB,JB)
      XA=ZF(Z1,Y1,1)+.0001
      XA=ZF(Z1,Y1,1)
      XB=ZF(Z2,Y2,1)
      DO 178 JA=2,NA
      IF(XR(IA,JA).LE.XA)GO TO 178
      IF(XR(IA,JA).GE.XB)GO TO 176
      QR(JA,JC)=QR(JA,JC)+DR(IB,JB)*(XR(IA,JA)-XA)/(XB-XA)
      GO TO 178
176   QR(JA,JC)=QR(JA,JC)+DR(IB,JB)
178   CONTINUE
      KA=NA
      DO 180 JA=2,NA
      QR(KA,JC)=QR(KA,JC)-QR(KA-1,JC)
      KA=KA-1
180   CONTINUE
      DO 190 JA=2,NC
      KC=0
      QRX=.0001
      DO 182 JC=2,NC
      IF(QR(JA,JC).LE.QRX)GO TO 182
      KC=JC
      NQ(JA)=KC
      QRX=QR(JA,JC)
182   CONTINUE
      DO 184 JC=2,NC
184   QR(JA,JC)=QR(JA,JC)/QRX
      CA(JA)=CA(JA)/QRX
      DO 188 J=2,NA
      IF(J.EQ.JA)GO TO 188
      QX=QR(J,KC)
      DO 186 JC=2,NC
      QR(J,JC)=QR(J,JC)-QX*QR(JA,JC)
186   CONTINUE
      CA(J)=CA(J)-QX*CA(JA)
188   CONTINUE
190   CONTINUE
      DO 192 JA=2,NC
      KC=NQ(JA)
192   CR(KC)=CA(JA)
      CR(1)=0.
      DO 194 JA=2,NC
      IF(CR(JA).LT.0.)CR(JA)=0.
194   CR(JA)=CR(JA)+CR(JA-1)
      GO TO 300
200   JA=1
```

```
      JB=1
      JC=0
      KA=0
      KB=0
      CA(1)=0.
      CB(1)=0.
      NM=NA
      IF(NM.LT.NB)NM=NB
205   IF(XR(IA,JA)-XR(IB,JB))210,235,255
210   IF(KA.GT.0)GO TO 260
      IF(XR(IA,JA).LT.XR(IB,1))GO TO 225
      DB=CB(JB-1)+(CB(JB)-CB(JB-1))*(XR(IA,JA)-XR(IB,JB-1))/
     .(XR(IB,JB)-XR(IB,JB-1))
      GO TO 220
215   DB=CB(NB)
220   JC=JC+1
      CR(JC)=CA(JA)*DB
      ZR(JC)=XR(IA,JA)
225   IF(JA.GE.NA)GO TO 230
      JA=JA+1
      CA(JA)=CA(JA-1)+DR(IA,JA)
      GO TO 205
230   IF(KB.GT.0)GO TO 280
      KA=1
      GO TO 205
235   JC=JC+1
      CR(JC)=CA(JA)*CB(JB)
      ZR(JC)=XR(IA,JA)
      IF(JA.GE.NA)GO TO 245
      JA=JA+1
      CA(JA)=CA(JA-1)+DR(IA,JA)
240   IF(JB.GE.NB)GO TO 250
      JB=JB+1
      CB(JB)=CB(JB-1)+DR(IB,JB)
      GO TO 205
245   IF(KB.GT.0)GO TO 280
      KA=1
      GO TO 240
250   IF(KA.GT.0)GO TO 280
      KB=1
      GO TO 205
255   IF(KB.GT.0)GO TO 215
      IF(XR(IB,JB).LT.XR(IA,1))GO TO 270
      DA=CA(JA-1)+(CA(JA)-CA(JA-1))*(XR(IB,JB)-XR(IA,JA-1))/
     .(XR(IA,JA)-XR(IA,JA-1))
      GO TO 265
260   DA=CA(NA)
265   JC=JC+1
      CR(JC)=CB(JB)*DA
      ZR(JC)=XR(IB,JB)
270   IF(JB.GE.NB)GO TO 275
      JB=JB+1
      CB(JB)=CB(JB-1)+DR(IB,JB)
      GO TO 205
275   IF(KA.GT.0)GO TO 280
      KB=1
      GO TO 205
280   NC=JC
290   IF(NC.LE.NM)GO TO 300
      DJ=1.
      JJ=2
```

```
      NC=NC-1
      DO 294 J=2,NC
      DS=(CR(J-1)+(CR(J+1)-CR(J-1))*(ZR(J)-ZR(J-1))/(ZR(J+1)-ZR(J-1))
     .-CR(J))**2
      IF(DS.GE.DJ)GO TO 294
      DJ=DS
      JJ=J
      IF(DJ.LT..0001)GO TO 296
  294 CONTINUE
  296 DO 298 J=JJ,NC
      CR(J)=CR(J+1)
  298 ZR(J)=ZR(J+1)
      GO TO 290
  300 F=1./CR(NC)
      DO 302 J=2,NC
  302 CR(J)=F*CR(J)
      XR(IC,1)=ZR(1)
  308 DO 310 J=2,NC
      DR(IC,J)=CR(J)-CR(J-1)
  310 XR(IC,J)=ZR(J)
      NX(IC)=NC
 1000 DR(IC,1)=AVF(IC)
      IF(K.EQ.5) RETURN
      IF(DR(IC,1).EQ.0.) RETURN
      F=CM/DR(IC,1)
      CALL HMS(IC,F,IC)
      RETURN
      END
```

```
      SUBROUTINE HAS(N)
C     HISTOGRAM ADJUSTING SUBROUTINE
      COMMON KK,IK,KR,RUN(7),KS,STG(7),KT,TSK(7),DSC(6),ID,
     .NE,AES(4),IEC(10),IET(10),AEC(10),AET(10),AED(10),AEF(10,4),
     .NO,IOT(20),IOQ(20),IOC(20),
     .NP,DAC(120,2),IU(120),IAB(120,6),AAG(120),AAS(120),AAT(120),
     .ND,NX(572),DR(572,11),XR(572,11),KO,
     .NS,KKT(20),IR(5),AR(4),APA(2),
     .NT,JT(100),XMT(100),XVT(100),L1,L2,L3,L4,KTS,ITE
      DIMENSION C(11),D(11),Z(11)
      M=NX(N)
      IF(M.EQ.1) RETURN
      IF(ID.LE.0) RETURN
      IF(NX(ID).LE.0) RETURN
      AVER=AVF(N)
      MD=NX(ID)
      IF(MD.GT.1) GO TO 5
      XR(N,1)=AVER
      NX(N)=1
      RETURN
    5 C(1)=0.
      D(1)=0.
      Z(MD)=XR(N,M)
      DO 10 I=2,M
   10 C(I)=C(I-1)+DR(N,I)
      IF(C(M).EQ.1.) GO TO 19
      IF(C(M).EQ.0.) RETURN
      DO 15 I=2,M
   15 C(I)=C(I)/C(M)
   19 DO 20 I=2,MD
   20 D(I)=D(I-1)+DR(ID,I)
      IF(D(MD).EQ.1.) GO TO 27
      IF(D(MD).EQ.0.) RETURN
      DO 25 I=2,MD
      DR(ID,I)=DR(ID,I)/D(MD)
   25 D(I)=D(I)/D(MD)
   27 IF(D(MD-1).NE.1.) GO TO 30
      MD=MD-1
      NX(ID)=MD
      GO TO 27
   30 I=2
      J=2
   31 DR(N,I)=DR(ID,I)
   32 IF(C(J).LE.D(I)) GO TO 33
      Z(I)=XR(N,J-1)+(D(I)-C(J-1))*(XR(N,J)-XR(N,J-1))/(C(J)-C(J-1))
      I=I+1
      GO TO 31
   33 J=J+1
      IF(J.LE.M) GO TO 32
      DO 35 I=2,MD
   35 XR(N,I)=Z(I)
      NX(N)=MD
      AVR=AVF(N)
      IF(AVR.EQ.0.) RETURN
      F=AVER/AVR
      DO 40 I=1,M
   40 XR(N,I)=XR(N,I)*F
      RETURN
      END
```

```
      SUBROUTINE HES(J,K)
C
C     HISTOGRAM EQUATING SUBROUTINE
C     SET J=K.
      COMMON KK,IK,KR,RUN(7),KS,STG(7),KT,TSK(7),DSC(6),ID,
     .NE,AES(4),IEC(10),IET(10),AEC(10),AET(10),AED(10),AEF(10,4),
     .NO,IOT(20),IOQ(20),ICC(20),
     .NP,DAC(120,2),IU(120),IAB(120,6),AAG(120),AAS(120),AAT(120),
     .ND,NX(572),DR(572,11),XR(572,11),KO,
     .NS,KKT(20),IR(5),AR(4),APA(2),
     .NT,JT(100),XMT(100),XVT(100),L1,L2,L3,L4,KTS,ITE
      M=NX(K)
      NX(J)=M
      DO 10 I=1,M
      DR(J,I)=DR(K,I)
   10 XR(J,I)=XR(K,I)
      RETURN
      END
```

```
      SUBROUTINE HMS(IA,C,IC)
C
C     HISTOGRAM MULTIPLYING SUBROUTINE
C     HISTOGRAM IC EQUALS IA MULTIPLIED BY CONSTANT C
      COMMON KK,IK,KR,RUN(7),KS,STG(7),KT,TSK(7),DSC(6),ID,
     .NE,AES(4),IEC(10),IET(10),AEC(10),AET(10),AED(10),AEF(10,4),
     .NO,IOT(20),IOQ(20),IOC(20),
     .NP,DAC(120,2),IU(120),IAB(120,6),AAG(120),AAS(120),AAT(120),
     .ND,NX(572),DR(572,11),XR(572,11),KO,
     .NS,KKT(20),IR(5),AR(4),APA(2),
     .NT,JT(100),XMT(100),XVT(100),L1,L2,L3,L4,KTS,ITE
      IF(C.EQ.0.)GO TO 20
      M=NX(IA)
      NX(IC)=M
      DO 10 J=1,M
      DR(IC,J)=DR(IA,J)
   10 XR(IC,J)=XR(IA,J)*C
      DR(IC,1)=DR(IA,1)*C
      RETURN
   20 NX(IC)=1
      DR(IC,1)=0.
      XR(IC,1)=0.
      RETURN
      END
```

A37

```
      SUBROUTINE HWS(I)
C  HISTOGRAM WRITING ROUTINE
C  VERSION 16 JULY,1969
      COMMON KK,IK,KR,RUN(7),KS,STG(7),KT,TSK(7),DSC(6),ID,
     ,NE,AES(4),IEC(10),IET(10),AEC(10),AET(10),AED(10),AEF(10,4),
     ,NO,IOT(20),IOQ(20),IOC(20),
     ,NP,DAC(120,2),IU(120),IAB(120,6),AAG(120),AAS(120),AAT(120),
     ,ND,NX(572),DR(572,11),XR(572,11),KO,
     ,NS,KKT(20),IR(5),AR(4),APA(2),
     ,NT,JT(100),XMT(100),XVT(100),L1,L2,L3,L4,KTS,ITE
      M=NX(I)
      IF(M.LE.1) GO TO 100
      D=AVF(I)
      WRITE(6,600) D
  600 FORMAT(1H+,46X,13HMEAN VALUE = ,E12.5)
      WRITE(6,601) (DR(I,J),J=2,M)
  601 FORMAT(15H  PROBABILITY =,5X,10F10.5)
      WRITE(6,602) (XR(I,J),J=1,M)
  602 FORMAT(10H  RANGE = ,11E10.2)
      WRITE(6,609)
  609 FORMAT(/)
      RETURN
  100 WRITE(6,603) XR(I,1)
  603 FORMAT(1H+,46X,2H= ,E12.5/)
      RETURN
      END
```

```
      SUBROUTINE MAS(I,T)
C
C     MICROBIAL ACCRETION SUBROUTINE
C     I IS THE PART AFFECTED, T THE TIME OF UPDATE
      COMMON KK,IK,KR,RUN(7),KS,STG(7),KT,TSK(7),DSC(6),ID,
     .NE,AES(4),IEC(10),IET(10),AEC(10),AET(10),AED(10),AEF(10,4),
     .NO,IOT(20),IOQ(20),IOC(20),
     .NP,DAC(120,2),IU(120),IAB(120,6),AAG(120),AAS(120),AAT(120),
     .ND,NX(572),DR(572,11),XR(572,11),KO,
     .NS,KKT(20),IR(5),AR(4),APA(2),
     .NT,JT(100),XMT(100),XVT(100),L1,L2,L3,L4,KTS,ITE
      DT=T-AAT(I)
      IF(DT.LE.0.)RETURN
      AAT(I)=T
      IE=IAB(I,5)
      IF(IE.LE.0)RETURN
      IC=IEC(IE)
      I1=ND+1
      I2=ND+2
      I3=ND+3
      C=AAG(I)*AEC(IE)/DR(IC,1)
      DO 40 J=1,4
      IF(IAB(I,J).EQ.0)GO TO 40
      IF(AEF(IE,J).EQ.0.)GO TO 40
      IB=IAB(I,J)
      A=DR(IB,1)
      F=C*AEF(IE,J)*A
      CALL HMS(IC,F,I1)
      DR(I1,1)=AVF(I1)
      V=AET(IE)*AES(J)
      F=EXP(-DT/V)
      CALL HMS(IB,F,I2)
      DR(I2,1)=AVF(I2)
      F=V*(1.-F)
      CALL HMS(I1,F,I3)
      CALL HCS(I2,I3,IB,1)
      DR(IB,1)=A
   40 CONTINUE
      RETURN
      END
```

```
      FUNCTION AVF(I)
C     AVERAGE VALUE FUNCTION
      COMMON KK,IK,KR,RUN(7),KS,STG(7),KT,TSK(7),DSC(6),ID,
     .NE,AES(4),IEC(10),IET(10),AEC(10),AET(10),AED(10),AEF(10,4),
     .NO,IOT(20),IOQ(20),IOC(20),
     .NP,DAC(120,2),IU(120),IAB(120,6),AAG(120),AAS(120),AAT(120),
     .ND,NX(572),DR(572,11),XR(572,11),KO,
     .NS,KKT(20),IR(5),AR(4),APA(2),
     .NT,JT(100),XMT(100),XVT(100),L1,L2,L3,L4,KTS,ITE
      M=NX(I)
      IF(M.GT.1) GO TO 10
      AVF=XR(I,1)
      RETURN
   10 AVF=0.
      DO 20 J=2,M
   20 AVF=AVF+DR(I,J)*(XR(I,J-1)+XR(I,J))
      AVF=.5*AVF
      RETURN
      END
```

```
      FUNCTION ZF(X,Y,K)
C
C     Z FUNCTION OF X AND Y
C     K DETERMINES THE OPERATION + - * /.
      GO TO (10,20,30,40,50),K
   10 ZF=X+Y
      RETURN
   20 ZF=X-Y
      RETURN
   30 ZF=X*Y
      RETURN
   40 ZF=X/Y
      RETURN
   50 ZF=X
      IF(Y.GT.X)ZF=Y
      RETURN
      END
```

```
      SUBROUTINE UPS
C     DETAILED PRINTOUT SUBROUTINE  (READS FROM TAPE11)
      COMMON/X/M,MR,MG,MT,MS,DSC(4),N,NX,DR(11),XR(11),P(5),NP(5)
      REWIND 11
      READ(11) M,MR,MG,MT,MS,(DSC(J),J=1,4),N,NX,(DR(J),J=1,11),
     .(XR(J),J=1,11),(P(J),J=1,5),(NP(J),J=1,5)
      WRITE(6,601) MR,(DR(J),J=1,7)
  601 FORMAT(35H1MICROBIAL BURDEN DETAILED PRINTOUT/4H RUN,I2,2X,7A6//)
      READ(5,500) K,NS,NT,NST,NS2,NT2,NST2
  500 FORMAT(7I5)
C     K=0 CALLS EXIT
C     K=1 PRINTS STAGE NS,TASK NT (SUBTASK NST)
C     K=2 PRINTS FROM NS,NT,(NST) TO NS2,NT2,(NST2)
   10 READ(11) M,MR,MG,MT,MS,(DSC(J),J=1,4),N,NX,(DR(J),J=1,11),
     .(XR(J),J=1,11),(P(J),J=1,5),(NP(J),J=1,5)
      IF(M.LE.0) CALL EXIT
   11 IF(MG.NE.NS) GO TO 10
      IF(MT.NE.NT) GO TO 10
      IF((NST.GT.0).AND.(NST.NE.MS)) GO TO 10
      IF(K.EQ. 2) GO TO 20
   15 CALL PLS
      READ(11) M,MR,MG,MT,MS,(DSC(J),J=1,4),N,NX,(DR(J),J=1,11),
     .(XR(J),J=1,11),(P(J),J=1,5),(NP(J),J=1,5)
      IF(M.LE.0) CALL EXIT
      IF(MG.NE.NS) GO TO 100
      IF(MT.NE.NT) GO TO 100
      IF((NST.GT.0).AND.(NST.NE.MS)) GO TO 100
      GO TO 15
   20 NS=NS2
      NT=NT2
      NST=NST2
   25 CALL PLS
      READ(11) M,MR,MG,MT,MS,(DSC(J),J=1,4),N,NX,(DR(J),J=1,11),
     .(XR(J),J=1,11),(P(J),J=1,5),(NP(J),J=1,5)
      IF(M.LE.0) CALL EXIT
      IF(MG.NE.NS) GO TO 25
      IF(MT.NE.NT) GO TO 25
      IF((NST.GT.0).AND.(NST.NE.MS)) GO TO 25
  100 READ(5,500) K,NS,NT,NST,NS2,NT2,NST2
      IF(K.LE.0) CALL EXIT
      GO TO 11
      END
```

```
      SUBROUTINE PLS
C     PARAMETER LABELING SUBROUTINE
      COMMON/X/M,MR,MG,MT,MS,DSC(4),N,NX,DR(11),XR(11),P(5),NP(5)
  600 FORMAT(//40X,3HRUN,I2,7H, STAGE,I3,6H, TASK,I4,9H, SUBTASK,I3,2X,
     .4A6)
      GO TO(1,2,3,4,5,6,7,8,9),M
    1 WRITE(6,601) MR,(DR(J),J=1,7),MG,(DR(J),J=8,11),(XR(J),J=1,3)
  601 FORMAT(35H1MICROBIAL BURDEN DETAILED PRINTOUT/4H RUN,I2,2X,7A6/
     .6H STAGE,I3,2X,7A6//)
      RETURN
    2 WRITE(6,600) MR,MG,MT,MS,(DSC(J),J=1,4)
      WRITE(6,602) NP(1),NP(2)
  602 FORMAT(31H+---- START OF NEW SUBTASK ----/23H PREREQUISITE SUBTASK
     .S-,I4,1H,I4,52X,10HSTART TIME)
      CALL HPS
      RETURN
    3 WRITE(6,600) MR,MG,MT,MS,(DSC(J),J=1,4)
      WRITE(6,603) NP(1),NP(2),P(1),NP(3),P(3),NP(4),P(4)
  603 FORMAT(26H+PART BURDEN AFTER FALLOUT/5H PART,I4,7H, SURF.,I2,
     .7H, AREA=,F7.2,7H, OPER.,I3,1H,,F6.2,6H HOURS,
     .10H, ENVIRON.,I3,1H,,F4.0,5H MEN.,10X,6HBURDEN)
      CALL HPS
      RETURN
    4 WRITE(6,600) MR,MG,MT,MS,(DSC(J),J=1,4)
      WRITE(6,604) NP(1),NP(2),(P(J),J=1,5)
  604 FORMAT(26H+PART BURDEN AFTER CONTACT/5H PART,I4,7H, SURF.,I2,
     .7H, AREA=,F7.2,10H, CONTACT=,F5.3,5H, S1=,F5.3,5H, S2=,F5.3,
     .7H, TOOL=,F7.1,7X,6HBURDEN)
      CALL HPS
      RETURN
    5 WRITE(6,600) MR,MG,MT,MS,(DSC(J),J=1,4)
      CALL HPS
      RETURN
    6 WRITE(6,600) MR,MG,MT,MS,(DSC(J),J=1,4)
      RETURN
    7 WRITE(6,600) MR,MG,MT,MS,(DSC(J),J=1,4)
      RETURN
    8 WRITE(6,600) MR,MG,MT,MS,(DSC(J),J=1,4)
      WRITE(6,608) NP(1),(P(J),J=3,4),NP(2),P(1),P(2),NP(4)
  608 FORMAT(25H+PART BURDEN AFTER UPDATE/5H PART,I4,2X,2A6,7H, SURF.,
     .I2,7H, AREA=,F7.2,7H, TIME=,F8.3,10H, ENVIRON.,I3,14X,6HBURDEN)
      CALL HPS
      RETURN
    9 WRITE(6,600) MR,MG,MT,MS,(DSC(J),J=1,4)
      WRITE(6,609) NP(1),NP(2),P(1),NP(3),P(4),P(5)
  609 FORMAT(34H+PART BURDEN AFTER DECONTAMINATION/5H PART,I4,7H, SURF.,
     .I2,7H, AREA=,F7.2,7H, OPER.,I3,1H,,F6.2,25H HOURS, FRACTION REMOVE
     .D=,F6.3,8X,6HBURDEN)                                              A
      CALL HPS
      RETURN
      END
```

```
      SUBROUTINE HPS
C   HISTOGRAM WRITING ROUTINE
      COMMON/X/M,MR,MG,MT,MS,DSC(4),N,NX,DR(11),XR(11),P(5),NP(5)
      IF(NX.LE.1) GO TO 100
      D=0.
      DO 20 J=2,NX
   20 D=D+.5*DR(J)*(XR(J-1)+XR(J))
      WRITE(6,600) D
  600 FORMAT(1H+,94X,13HMEAN VALUE = ,E12.5)
      WRITE(6,601) (DR(J),J=2,NX)
  601 FORMAT(15H  PROBABILITY =,5X,10F10.5)
      WRITE(6,602) (XR(J),J=1,NX)
  602 FORMAT(10H  RANGE = ,11E10.2)
      RETURN
  100 WRITE(6,603) XR(1)
  603 FORMAT(1H+,94X,2H= ,E12.5)
      RETURN
      END
```